MSX 2+

Panasonic

BASIC入門

- BASIC編
- MSX-DOS編
- 資料編



この本でBASICやMSX-DOSを勉強されるかたへ

この本は、初めてパーソナルコンピュータ（以降、「パソコン」）に触れるかたのために、ＭＳＸ２+パソコンに搭載されているプログラミング言語「漢字ＢＡＳＩＣ」とＭＳＸパソコンの標準オペレーティングシステム「ＭＳＸ－ＤＯＳ」の基本的なことがらを説明しています。

この本をお読みになるかたに、次のことを守ってくださるようお願いします。

- ・先に、パソコンに付属している「取扱説明書」をお読みください。
- ・前のページから順番にお読みください。
- ・できるだけ、説明のとおりにパソコンを操作しながらお読みください。
- ・途中でわからなくなったら、前のほうの説明から読み直してみてください。

※この本では一部の命令や関数についての説明を省略しています。
この本に書いてあることがらがひと通り分かるようになりましたら、書店やパソコンの販売店などで販売されているＭＳＸパソコン用のＢＡＳＩＣの解説書などをお読みになるようお勧めします。

表記上の約束

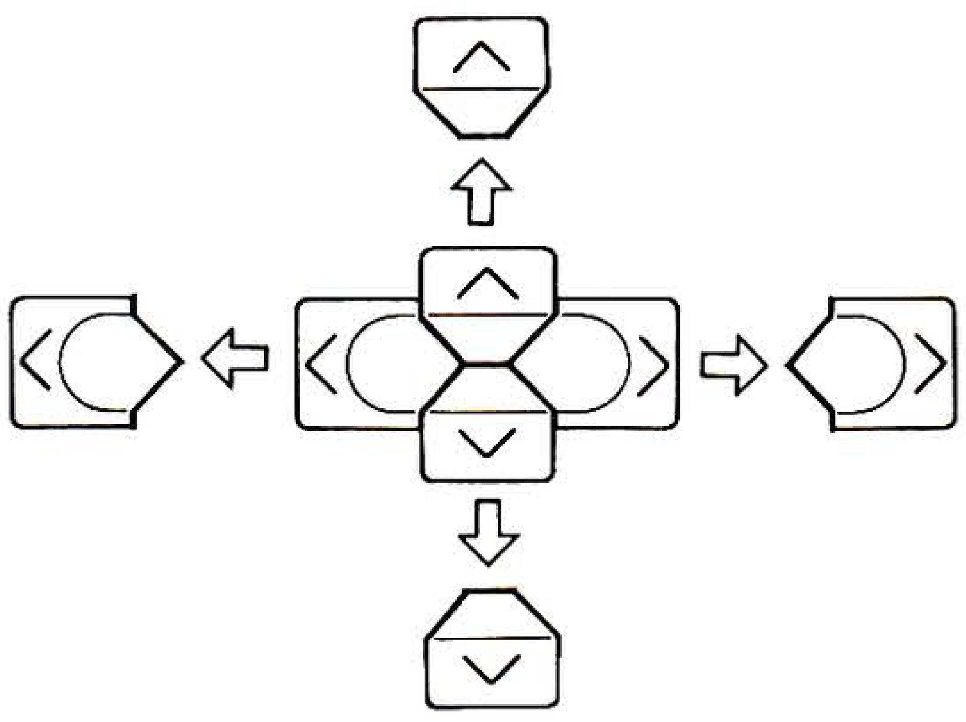
この本は次のような約束にしたがって書かれています。最初に覚えてください。

●説明の中の記号の意味は

ご注意	特に注意して欲しいことを説明します。 必ず読んでください。
◆	特定の場合にだけあてはまる説明をします。 ご自分にあてはまるところだけ読んでください。
メモ ※	各項目の補足説明や、知っておくと便利なことがらを説明しています。
（⇨●●ページ）	参照ページを示します。

●キーの表記（書き表しかた）は

・カーソルキーは次のように表記します。



・必要な文字だけを表示する場合は、その文字だけを□で囲んで表記します。



・表の中などでは、あるキーを押しながら別のキーを押すことを、2つのキーを＋記号でつないで表記する場合があります。

例）　CTRL ＋ STOP

CTRL キーを押しながら STOP キーを押す意味です。

・□は SPACE をあらわします。

●下のようなマークは

フロッピーディスクドライブのマーク ‥‥ フロッピーディスクドライブ　　データレコーダのマーク ‥‥ データレコーダ

プリンタのマーク ‥‥ プリンタ　　漢 ‥‥ 漢字プリンタ

その部分の説明のとおりに操作を行うのに必要な機器をあらわします。

BASIC編

MSX-DOS編

資　料　編

もくじ

ＢＡＳＩＣ編

ＭＳＸ－ＤＯＳ編

資　料　編

パソコンって何だろう?

パソコンを使い始める前に、まずパソコンとはどんなものなのか、
他の電気製品とはどんなところが違うのかを理解しておきましょう。

パソコンの特徴は

パソコンとテレビや冷蔵庫のような普通の電気製品とのあいだには、次のような違いがあります。

1．使いみちを決めるのは

テレビや冷蔵庫の使いみちはだれでも知っています。まさかテレビでスイカを冷やそうとしたり、冷蔵庫で「今週のベストテン」を見ようとする人はいないでしょう。
これらの「電気製品」は、工場で作られたときから使いみちが決められているのです。
パソコンは少し違います。工場では使いみちを決めないで作られていますし、お店の人も「これは○○に使うものだ」と決めて売ったりはしていません。
パソコンを何に使うのかは、あなた自身が決めることなのです。

2．目的によって使い分けるには

テレビや冷蔵庫なら、最初に正しく接続してしまえば、あとはスイッチを入れるだけで働き始めます。パソコンは違います。接続して電源スイッチを入れるだけでは何もしてくれません。
どんな仕事をしたらよいのかをまだパソコンに指示していないからです。
こういうと、「いや、そんなことはない。ぼくのパソコンはスイッチを入れるだけでテレビにきれいな画面が映るよ」という人もいるでしょう。たしかにパソコンの機種によっては、そのような動きかたをするものもあります。でもそれは、電源が入ったときにはそういう仕事をしろという指示が、あらかじめ工場でパソコンにあたえられているからです。

パソコン本体やそれに接続して使う各種の機器は、目で見ることや手で触ることができます。
これに対してパソコンに仕事をさせる指示は、見ることも触ることもできません。しかし、パソコンに何かの働きをさせるには、どちらも同じように大切なものなのです。

そこで、パソコン本体やその他の機器のことを**「ハードウェア」**（**「ハード」**）と呼び、パソコンにあたえる指示などのことを**「ソフトウェア」**（**「ソフト」**）と呼んで対等に扱います。

パソコンが実際にどんな働きかたをするかは、どんなソフトをあたえるかによって決まるのです。

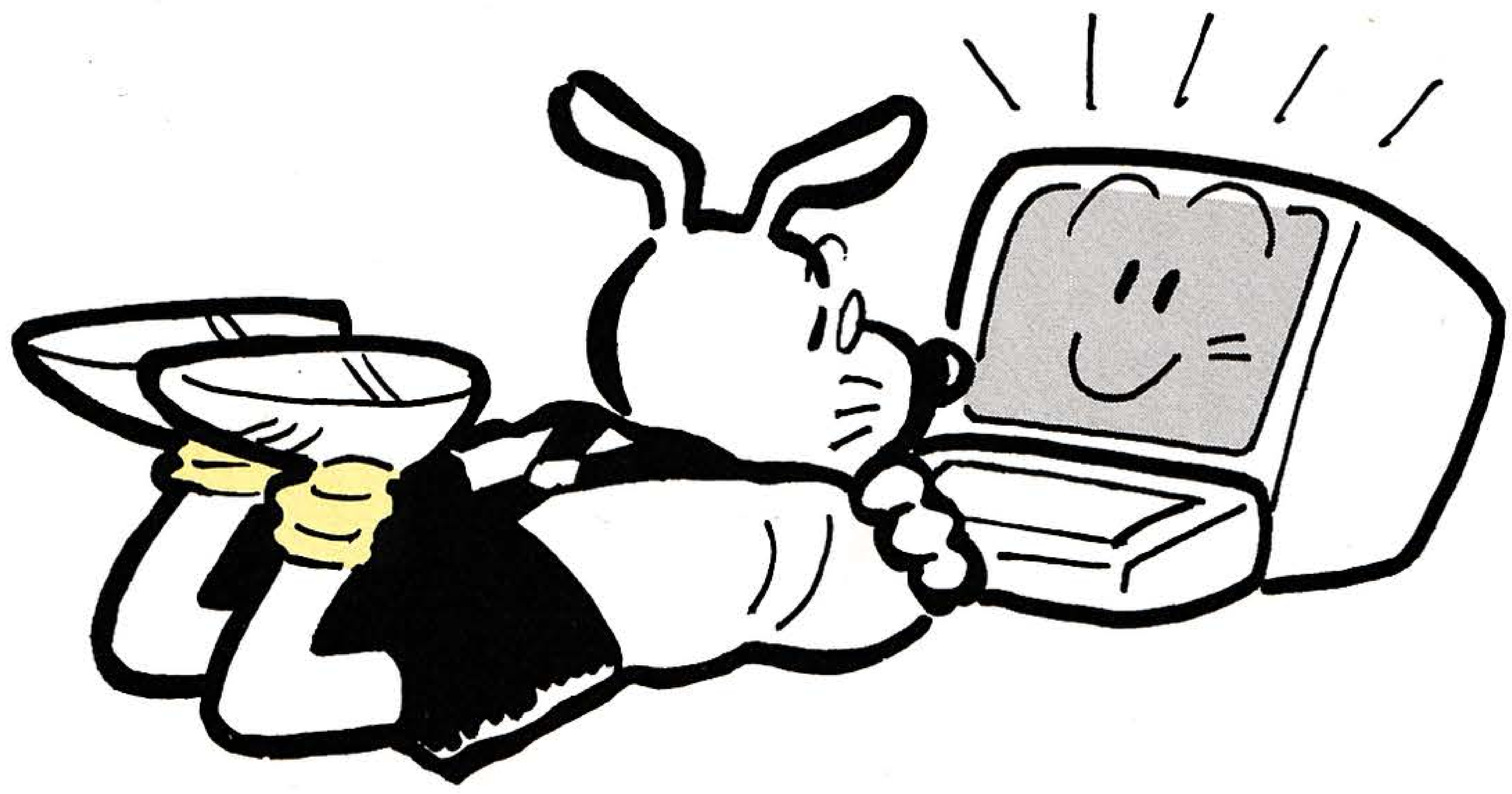
「BASIC」って何だろう?

この本の表紙にもある「ＢＡＳＩＣ」（ベーシック）というのは、いったい何でしょうか？　また、「ＭＳＸ２＋」（エムエスエックス・ツー・プラス）とは、何をあらわすのでしょうか？　ここではその疑問に答えます。

「BASIC」とは

パソコンを動かすソフトはいろいろな方法で手に入れることができます。ゲームやワープロ（ワードプロセッサ）としてお店で売られているソフトもありますし、雑誌に掲載されることもあります。また、ソフトを自分で作ることもできます。

ソフトを自分で作るということは、パソコンに自分で命令をあたえるということです。でも、パソコンはただの機械ですから、人間のことばは分かりません。
そこで、機械であるパソコンと人間の両方に分かることばが必要になります。
人間がそのことばでパソコンに命令すれば、パソコンはあたえられた命令どおりに働くわけです。
このようなことばを**「プログラミング言語」**と呼びます。

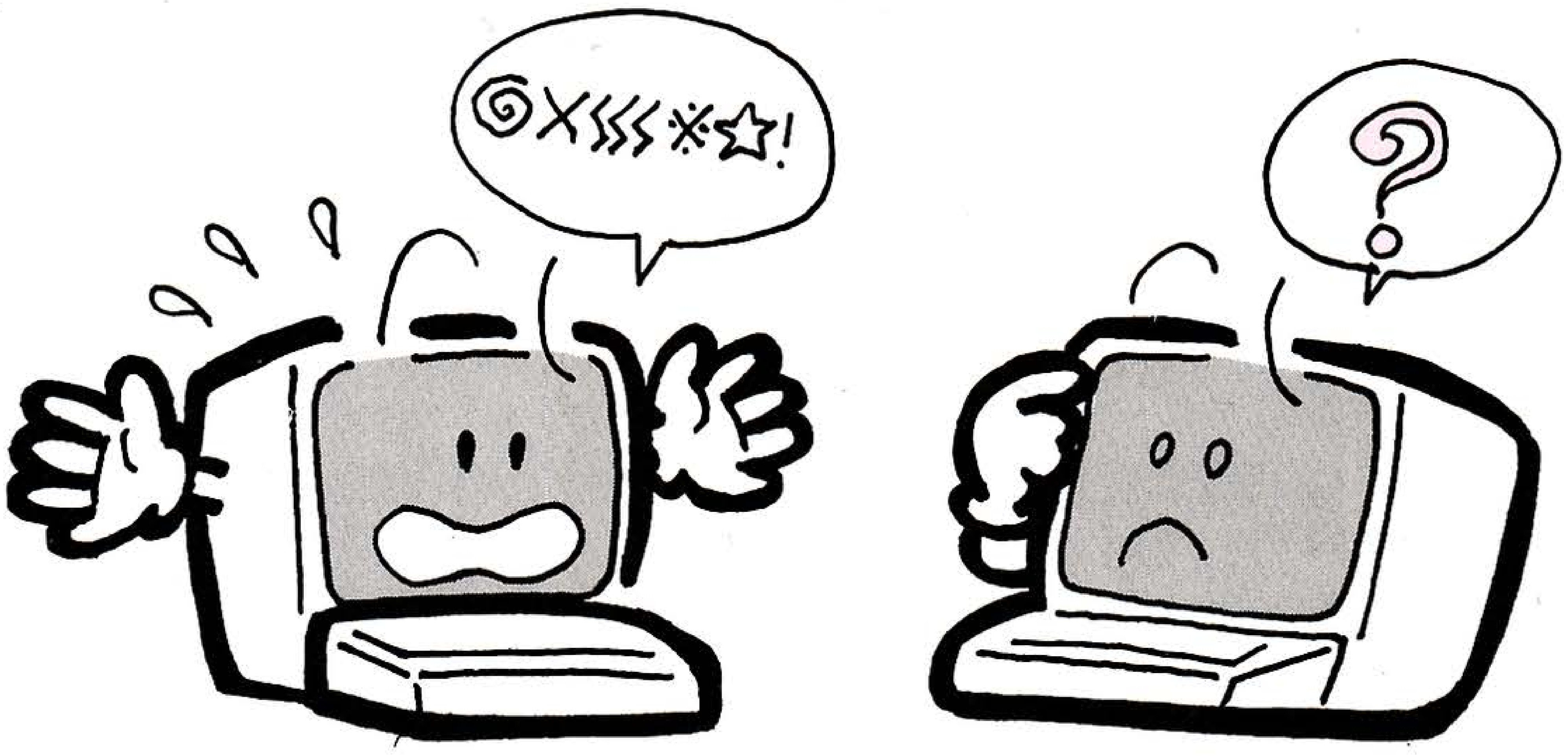


ＢＡＳＩＣはそのプログラミング言語のひとつです。
ＢＡＳＩＣの他にもたくさんのプログラミング言語が作られましたが、初めてプログラミング言語を学ぶ人にとっては、ＢＡＳＩＣが一番覚えやすくて親切なことばであるため、ほとんどのパソコンはＢＡＳＩＣで命令できるようになっています。

「MSX2+」とは

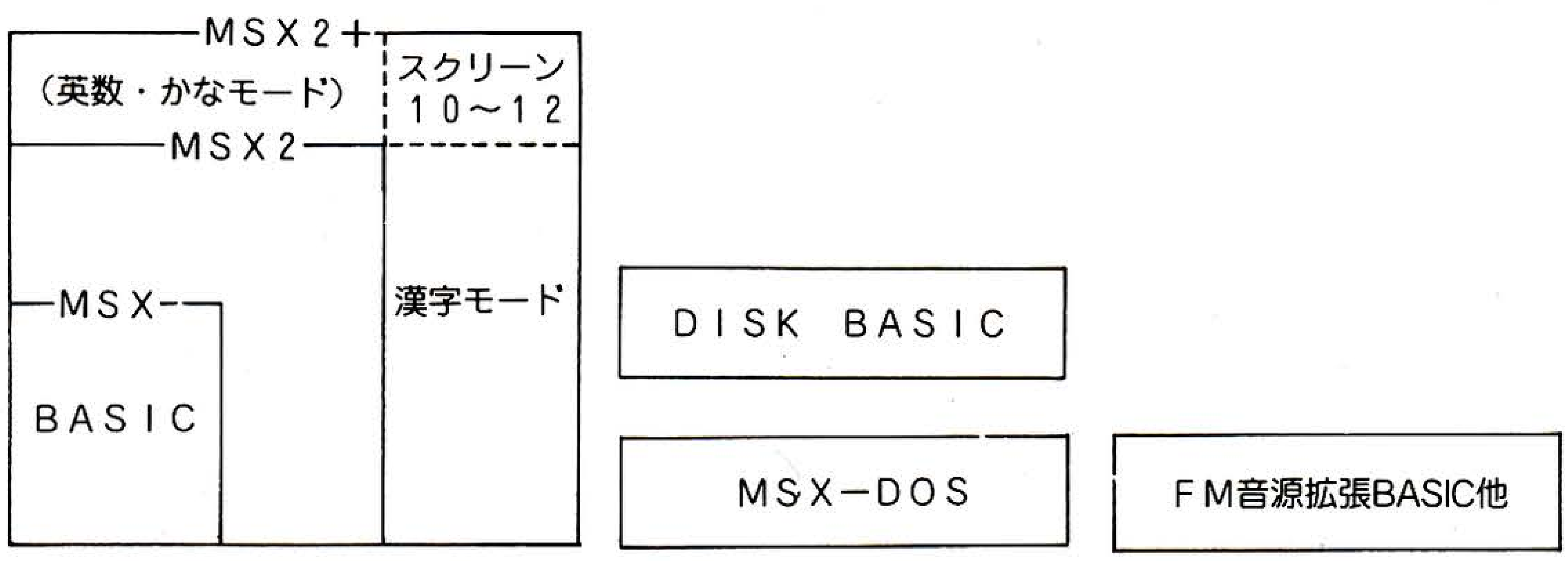
では、いろいろなパソコンのＢＡＳＩＣで作ったソフトを、おたがいに交換して使うことはできないのでしょうか？

結論から先に言うと、ほとんどの場合は、そのままでは使えません。なぜなら、パソコンの機種によって、ＢＡＳＩＣの命令の種類や文法（ことばの決まり）に少し違いがあるからです。



あなたのパソコンは、「ＭＳＸ２+ パソコン」と呼ばれるものです。ＭＳＸ２+ パソコンにもいくつかの機種がありますが、すべて「ＭＳＸ２+ 」という規格（共通の約束ごと）にしたがって作られています。ですからＭＳＸ２+ パソコンどうしなら、ＢＡＳＩＣで作ったソフトをお互いに交換して使うことができます。

ＭＳＸ２+ 規格は、その前に作られた「ＭＳＸ２」、「ＭＳＸ」という２つの規格を、その中に含んでいます。

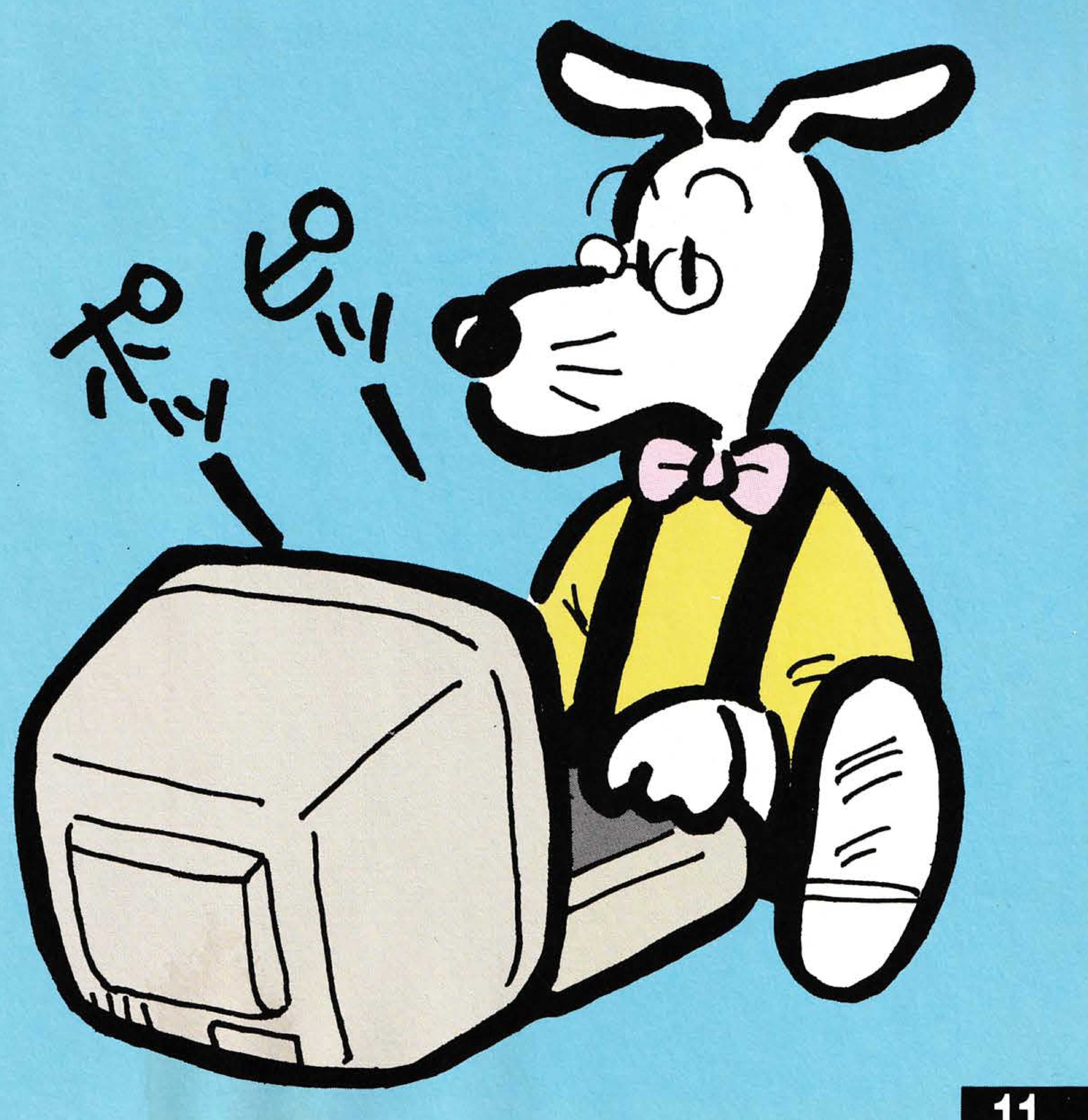


ですから、ＭＳＸ２+ パソコンでは、ＭＳＸパソコンやＭＳＸ２パソコンのために作られたソフトも使うことができます。
この本では、ＭＳＸ２+ パソコン用のＢＡＳＩＣである「漢字ＢＡＳＩＣ」とＭＳＸ－ＤＯＳの基本的な使いかたを説明しています。

第1章

プログラムを自分で作れば、今日からあなたもパソコンのエキスパートです。

初めてのプログラム



BASICのスタート

ＢＡＳＩＣのスタートのしかたは、パソコンの機種により異なります。
パソコンの取扱説明書を読んで確かめてください。

◆パソコンの機種によっては、電源スイッチを入れる前に「内蔵ソフト切替スイッチ」などを操作することが必要です。

接続とスタートが正しくできたら、しばらくしてテレビの画面に下の文字が表示されます。

```
MSX Basic version 3.0
Copyright 1988 by Microsoft
xxxxx Bytes free
Disk basic version 1.0
Ok
█
```

← この行は機種によって表示されないことがあります。（Disk basic version 1.0 の行）

画面に「Ｏｋ」という文字が、その下に四角いマーク（カーソル）が出ていればＢＡＳＩＣが正常にスタートしています。

◆**いつまでも上の表示が画面に出てこなかったり、何も画面に表示されないときは、接続のしかたかＢＡＳＩＣのスタートのしかたをまちがえていますので、一度スイッチを切り、パソコンの取扱説明書を読み直しながら、接続とＢＡＳＩＣのスタートをやり直してください。**

BASICの終了のしかた

ＢＡＳＩＣの終了のしかたは、カンタンです。
ふつうは、ただパソコンの電源スイッチを切るかリセットボタンを押せば、
それでＢＡＳＩＣは終了します。
ただし、次の場合はちょっと待ってください。

アクセスランプが点灯しているとき

フロッピーディスクドライブのアクセス（または ＩＮ　ＵＳＥ）ランプが点灯しているときは、ランプが消えるまで終了しないでください。

プログラムを作ったとき

パソコンはポータブルのワープロなどとは異なり、電源スイッチを切ったりリセットボタンを押すと、入力したものがすべて消えてしまいます。消えては困る大切なプログラムなどを入力したときは、電源スイッチを切る前に保存しておきましょう。
プログラムの保存のしかたは、34ページから説明しています。

キーボードを使いこなそう

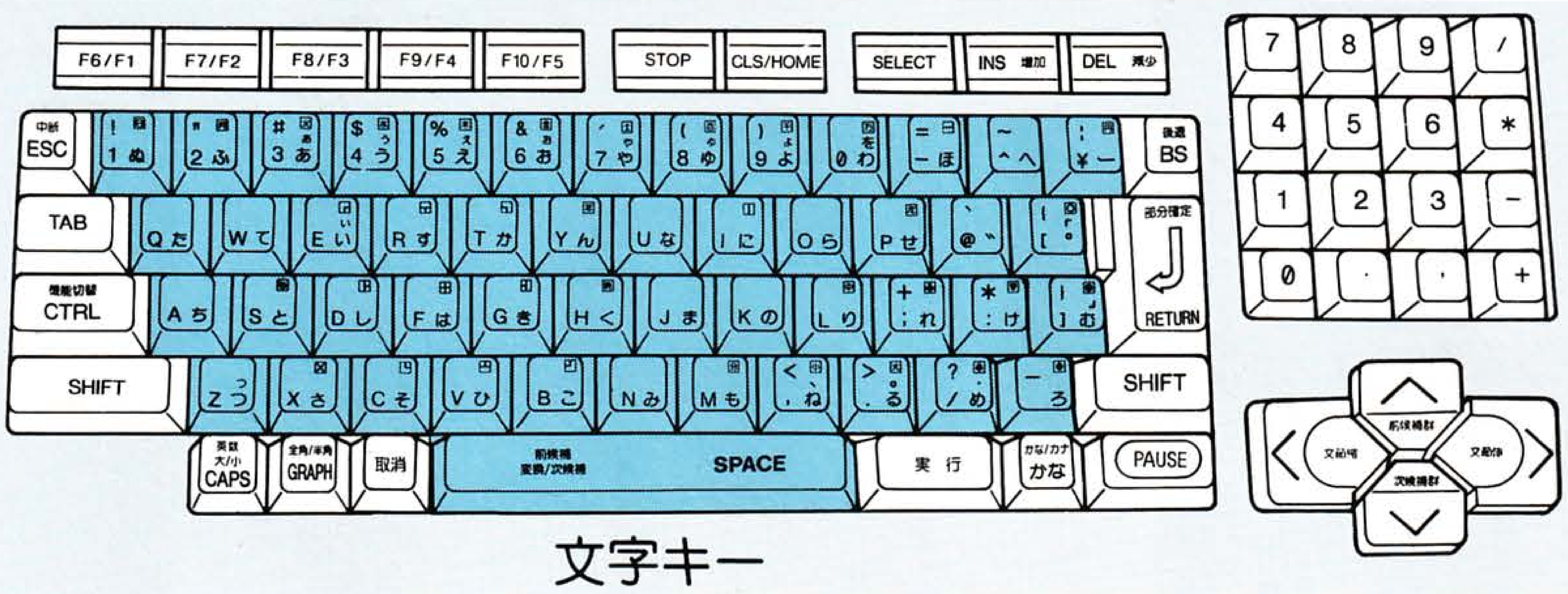
パソコンに命令をあたえるには、ふつうキーボードを使います。ですから、キーボードをうまく使えないうちは、パソコンをうまく使えないことになります。
まずキーボードの使いかたをひととおり知っておきましょう。

文字キーと特殊キー

キーボードに触れるのは初めてだという人は、たぶん「こんなにたくさんのスイッチのようなものを操作しないと使えないのなら、使いかたを覚えるのはたいへんだろうな」と思われたでしょう。

たしかにキーボードにはたくさんのキーが並んでいます。でも、そのほとんどは、キーの上に書かれている文字を一文字だけ画面に表示する働きしかしません。ですから、ただの鉛筆やボールペンのようなものだと思ってください。
これらのキーを**「文字キー」**と呼びます。

例) キーボード図
(FS-A1WX)



文字キー

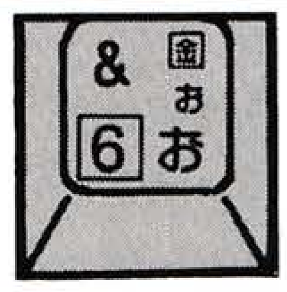
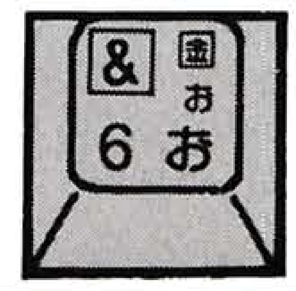
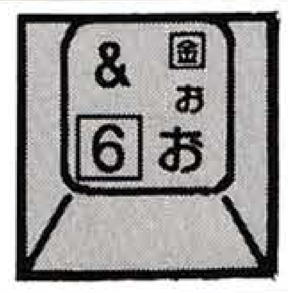
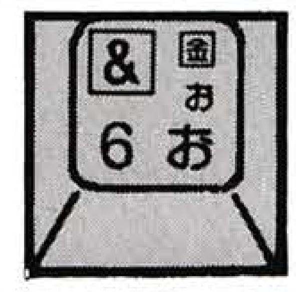
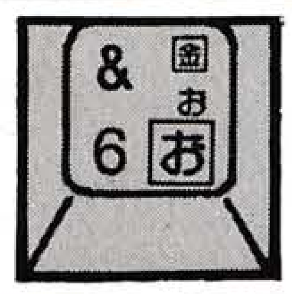
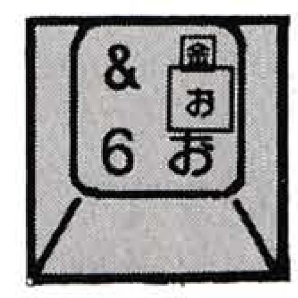
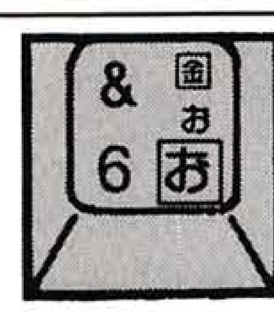
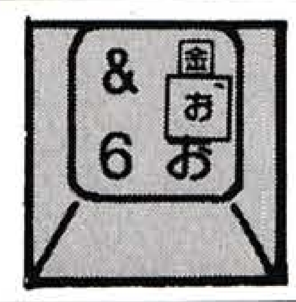
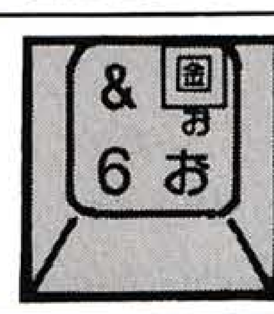
キーボードにある他のキーは、**「ファンクションキー」**とか**「特殊キー」**とか呼ばれるものでいろいろな機能をもっています。
その中で、下のキーだけは必ず覚えておいてください。

↵ (リターン)キー	……………………	表示した文字の入力
△▽◁▷ (カーソル)キー	……………………	表示位置の移動
SHIFT (シフト)キー	……………………	表示する文字の切替
かな キー	……………………	表示する文字の切替
GRAPH キー	……………………	表示する文字の切替

メモ: これ以降、特に理由がない限り、キーは「SHIFT」のように表記し、「キー」を省略します。

キーボードの使いかたの基本はパソコンの取扱説明書に書かれていますが、かんたんに整理すると、次ページの表のとおりです。

文字の表示のしかた

CAPSランプ	かなランプ	文字キーだけを押した時 表示される文字		SHIFTを押しながら文字キーを押した時 表示される文字	
消	消	英小文字、数字、英記号		英大文字、英記号	
点灯	消	英大文字、数字、英記号		英小文字、英記号	
消	点灯	ひらがな、かな記号		ひらがな、かな記号	
点灯	点灯	カタカナ、かな記号		カタカナ、かな記号	
GRAPH を押しながら		グラフィック文字			

※ CAPSランプ：CAPS を押すと点灯し、もう一度押すと消える。
　かなランプ　：かな を押すと点灯し、もう一度押すと消える。

※ SHIFT を押しながら かな を押してローマ字のつづりにしたがって英文字のキーを押すと、ひらがな（CAPSランプ消灯時）またはカタカナ（CAPSランプ点灯時）を画面に表示することができます。
（⇨198ページ）

BASICの2つの顔(ダイレクトモードとプログラムモード)

ＢＡＳＩＣの使いかたには、大きく分けて２種類あります。
ひとつは「ダイレクトモード」、もうひとつは「プログラムモード」です。
では、２つはどんなところが違うのでしょうか？

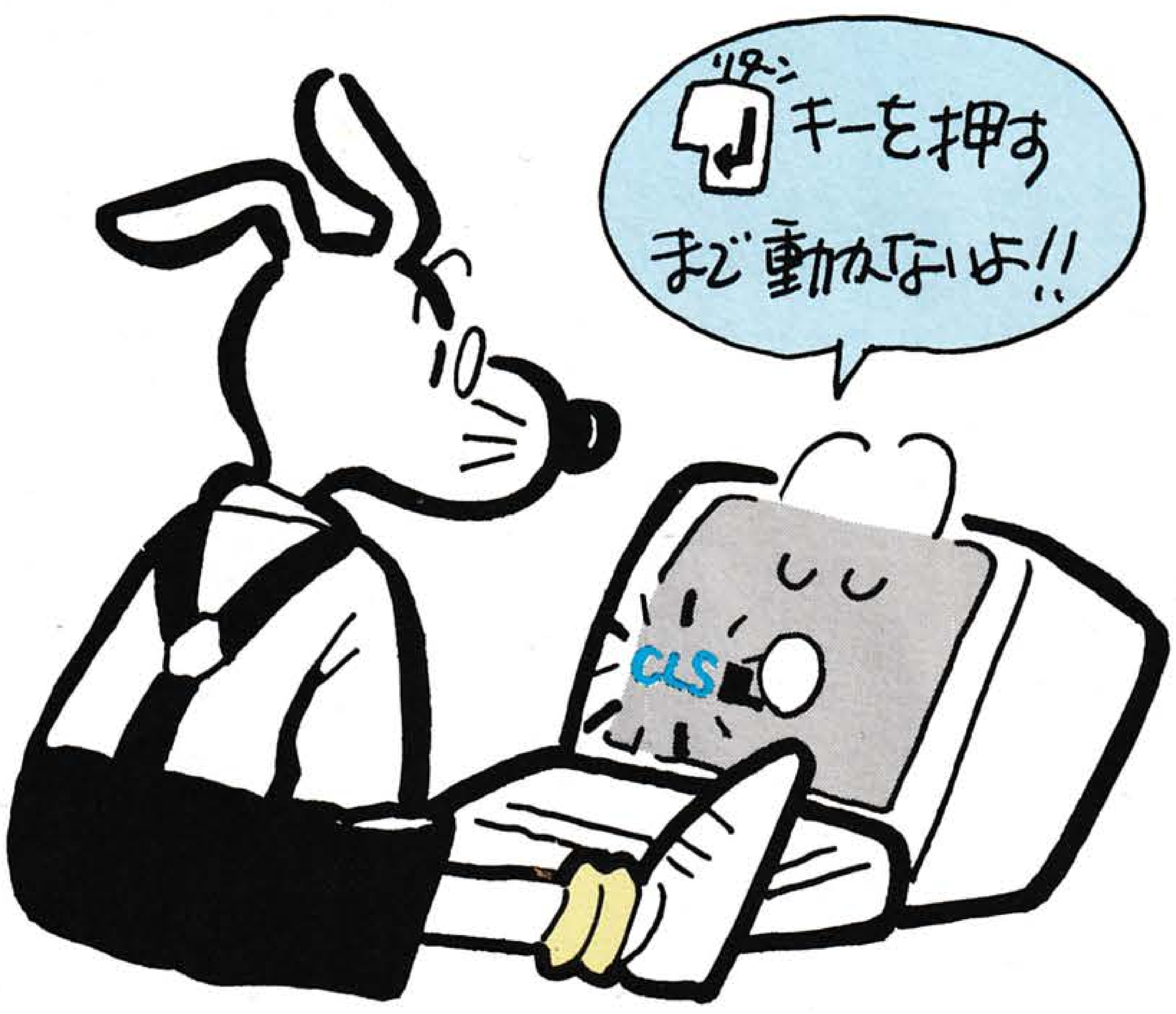
↵一発！ すぐ実行──ダイレクトモード

スタートしたときには、ＢＡＳＩＣはダイレクトモードになっています。ダイレクトモードでは、キーボードを使って人間がパソコンに直接命令します。命令するたびにすぐに実行されるので、ちょうど人間どうしが会話をするように命令することができます。
では、ダイレクトモードでパソコンに命令してみましょう。

「ＣＬＳ」（クリア・スクリーン）は、画面の表示をすべて消し、カーソルを画面の左上のすみに戻す命令です。
まず、次のようにキーをひと文字ずつ押してみてください。

C L S

押したキーの文字は画面に表示されますが、それ以上何も起こりません。
これは、パソコンがまだ命令を受け取っていないからです。



パソコンに命令を受け取らせるためには、文字キーを押して命令を画面に表示させたあと、↵ を押すことが必要です。

このとき、カーソルは表示された命令と同じ行になければいけません。
このように、命令に続けて ↵ を押し、パソコンに命令を受け取らせることを「命令を**入力する**」と呼びます。また、入力された命令にしたがって、パソコンが何かをすることを「命令を**実行する**」と呼びます。

では ⏎ を押してみましょう。画面は右のようになります。
これがＣＬＳ命令の実行を終わったときの状態です。

```
Ok
■




color auto  goto  list  run
```

命令の実行を終わると、パソコンはカーソルの位置に**「Ｏｋ」**と表示し、カーソルを次の行の左端に動かして次の命令を待ちます。

※ＢＡＳＩＣの命令は、小文字で入力しても同じように実行されます。

◆画面が上のようにならずに下の文字（シンタックス・エラー）が表示されたら、キーを押しまちがえていますから、もう一度正しく入力し直してください。

```
Syntax error
Ok
■
```

メモ：ＣＬＳ命令は、あとで説明する漢字テキスト画面では使えません。
漢字テキスト画面での画面の消去には、ＣＡＬＬ　ＣＬＳという命令を使用します。

まとめて実行――プログラムモード

ダイレクトモードでは、命令をひとつずつ入力して実行させます。しかし、パソコンに何かまとまった仕事をさせるときは、いくつもの命令が必要なので、ひとつずつ命令を入力しては実行させていたのではとてもたいへんです。

そこで、パソコンに実行させたい命令を順番に並べた**「プログラム」**というものを作ってあらかじめパソコンに入力し、入力が終わってからプログラムを実行させる命令を入力します。これが**「プログラム モード」**です。

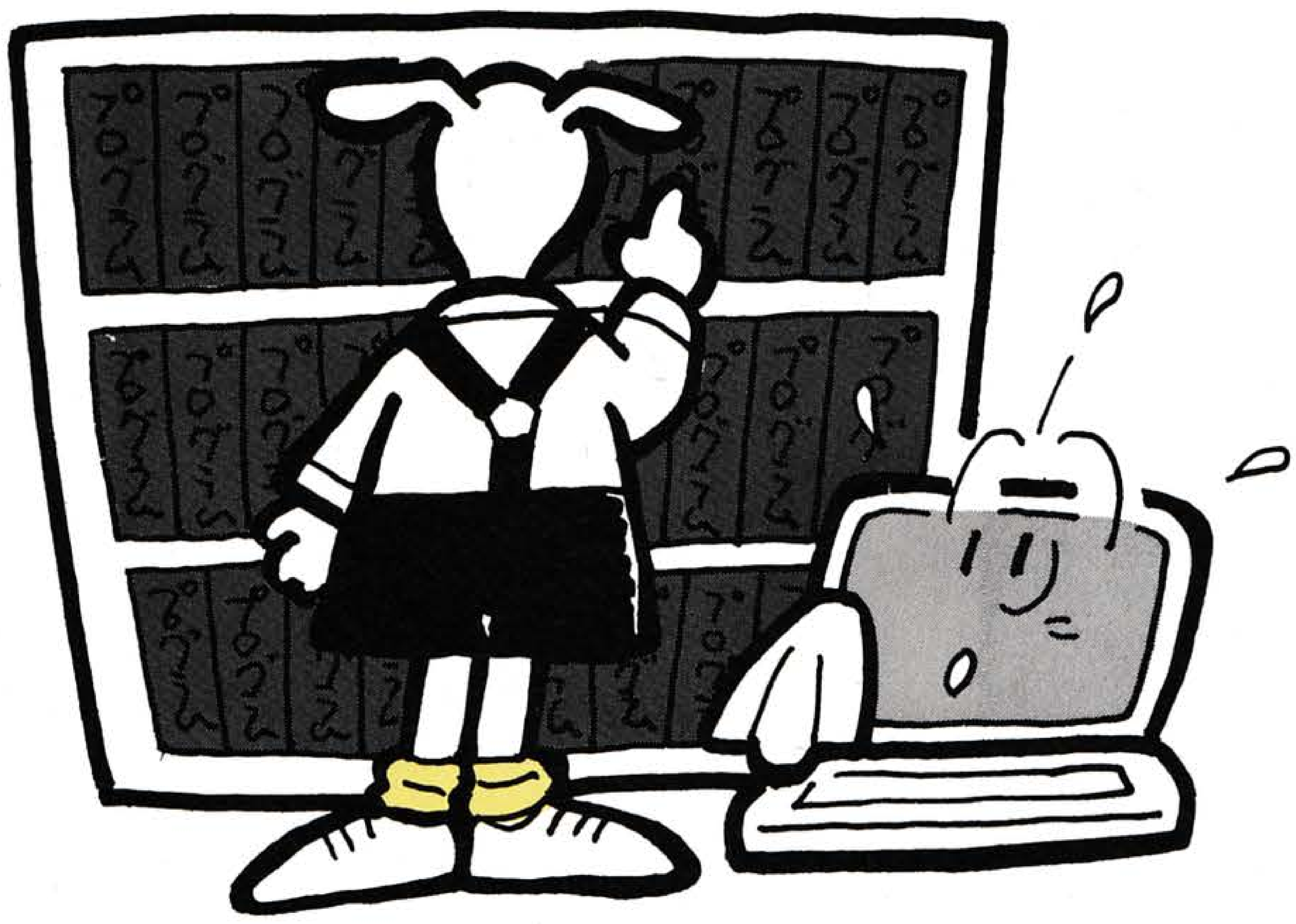


一度入力されたプログラムは、右のどれかの操作をするまで、パソコンが覚えています。

1．電源スイッチを切る
2．リセットボタンを押す
3．忘れるように命令する

ですから、プログラムを実行させる命令を入力するだけで、同じプログラムを何度でも繰り返し実行させることができます。

入力したプログラムは、フロッピーディスクに**「保存」**することができます。
保存したプログラムを次に使うときは、あらためてキーボードから入力する必要がありません。
保存したプログラムをパソコンの内部に**「呼び出し」**することにより、再び実行させることができるようになります。
（プログラムの保存と呼び出しについては、 34 ページから説明しています。）



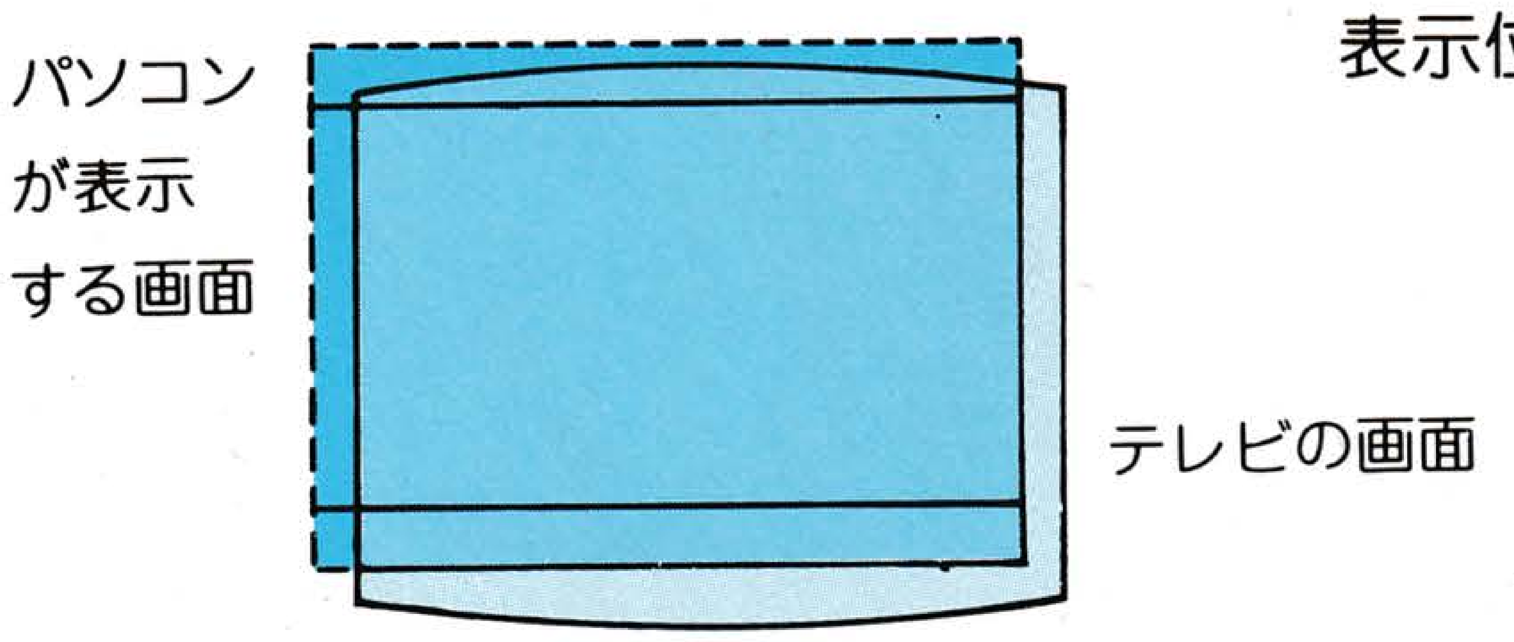
パソコンに直接命令してみよう(ダイレクトモード)

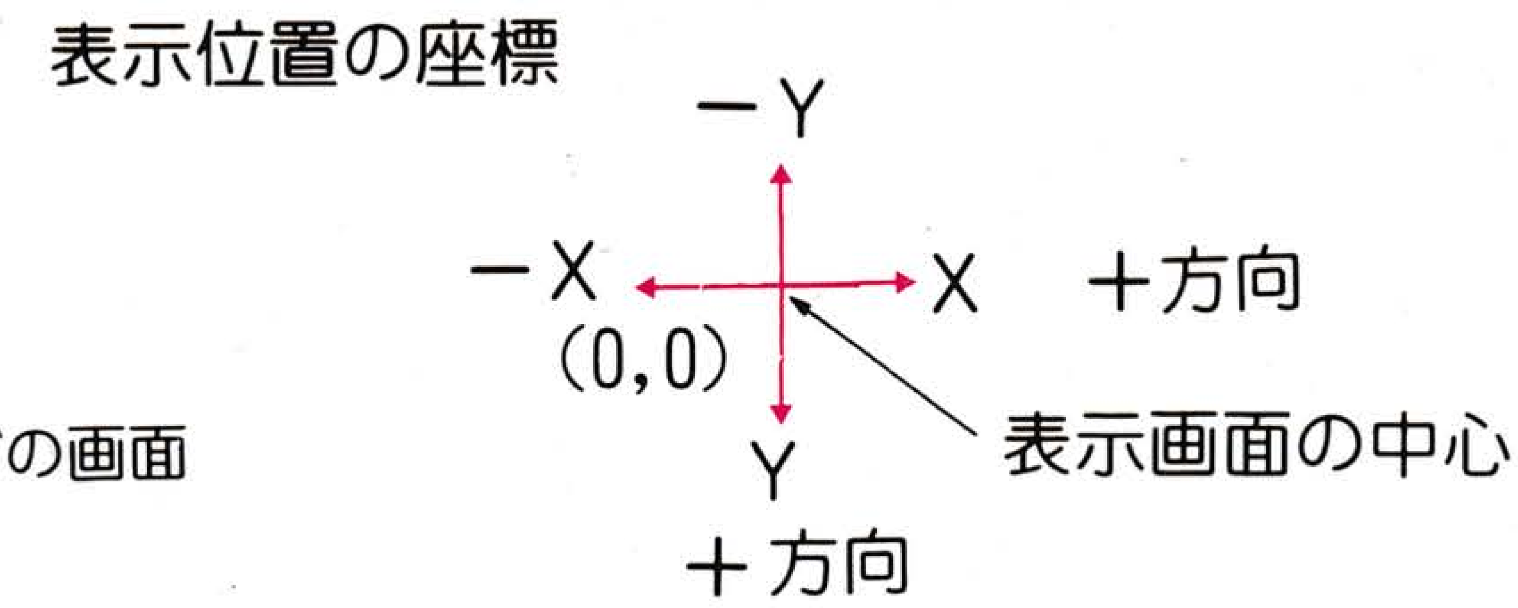
プログラムモードでパソコンを使う場合でも、
プログラムの入力や修正、実行や保存などの操作はダイレクトモードで行います。
ダイレクトモードでのパソコンの操作をもう少し練習しましょう。

画面の位置を合わす(SET ADJUST)

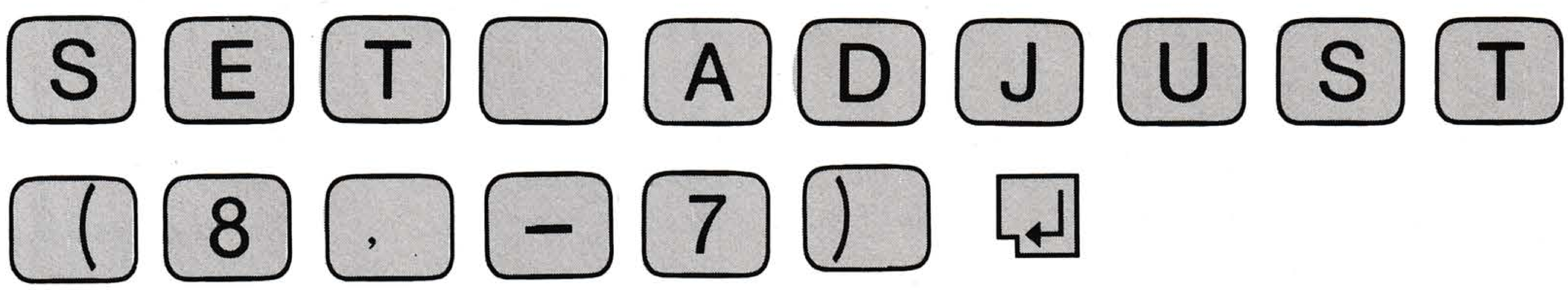
「SET　ADJUST」は、画面の表示位置を変更する命令です。パソコンの表示する画面が、ブラウン管の真ん中ではなく一方にずれて表示されるときに使用します。

BASICが初めてスタートしたときには、画面は（0，0）の位置に表示されます。





次のように入力して見ましょう。
（□は SPACE を、－ はマイナスをあらわします。）



SET　ADJUST（X，Y）

（－7～＋8の数字）
※小数点以下は無視されます。

画面が右上に動いたのがわかりましたか？
SET　ADJUST命令は、左のような基本の形で入力します。

- **Xの部分には、**
  画面の左右方向の位置をあらわす数字を入れます。
- **Yの部分には、**
  画面を上下方向の位置をあらわす数字を入れます。

画面を初めの位置に戻すには、SET　ADJUST（0，0）↵ と入力してください。

Xの部分を現在より大きい数字にすると右に、小さい数字にすると左に画面が移動します。	Yの部分を現在より大きい数字にすると下に、小さい数字にすると上に画面が移動します。

X、Yの部分の数字を変えて、パソコンの表示する画面がテレビのブラウン管の真ん中にくるように調整しておきましょう。これで、次にBASICをスタートしたときにも、設定した位置に画面が表示されます。（ただし、パソコンの乾電池が消耗したときや乾電池を入れ替えたときは、もう一度設定し直してください。）

画面に計算の答えや文字を表示する(PRINT)

PRINT（プリント）は、画面に文字や数字を表示させる命令です。
次のように入力してみましょう。

[P] [R] [I] [N] [T] [ ] [2] [+] [3] [↵]

```
PRINT  2+3
  5
Ok
■
```

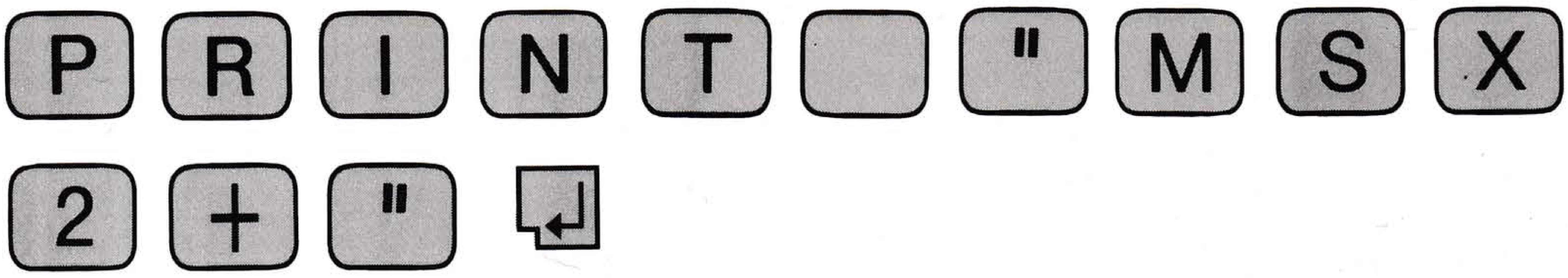
画面には左のように表示されますね。
PRINT命令の後ろに計算の式を書くと、計算の答えを画面に表示します。

BASICで計算に使う記号**（演算子）**の中には、学校などで習う記号とは形の違うものがあります。よく使われる演算子の書きかたを下の表にまとめておきます。

学校で教わる演算子	＋	−	×	÷	( )	X^2	例　((2+3)×5-6²)÷3
BASICの演算子	＋	−	*	/	( )	X^2	例　((2+3)*5-6^2)/3

上の表以外の演算子は、152ページに載っています。

画面に文字を表示するときは、表示したい文字を**「"」（ダブルクォート）**で囲みます。
このような、ひと続きの文字を**「文字列」**と呼びます。
次のように入力してください。




```
PRINT "MSX2+"
MSX2+
Ok
■
```

画面には、次のように表示されます。
「"」（ダブルクォート）で囲んだ文字列が、そのまま表示されましたね。

メモ：「"」で囲んだ文字は、たとえ数字や式であっても、文字列としてあつかわれます。

《ちょっとひと言》

命令のつづりをまちがえて入力したときなどに、下のような文字が画面に表示されて、まちがえたことを教えてくれます。

```
Syntax error
```

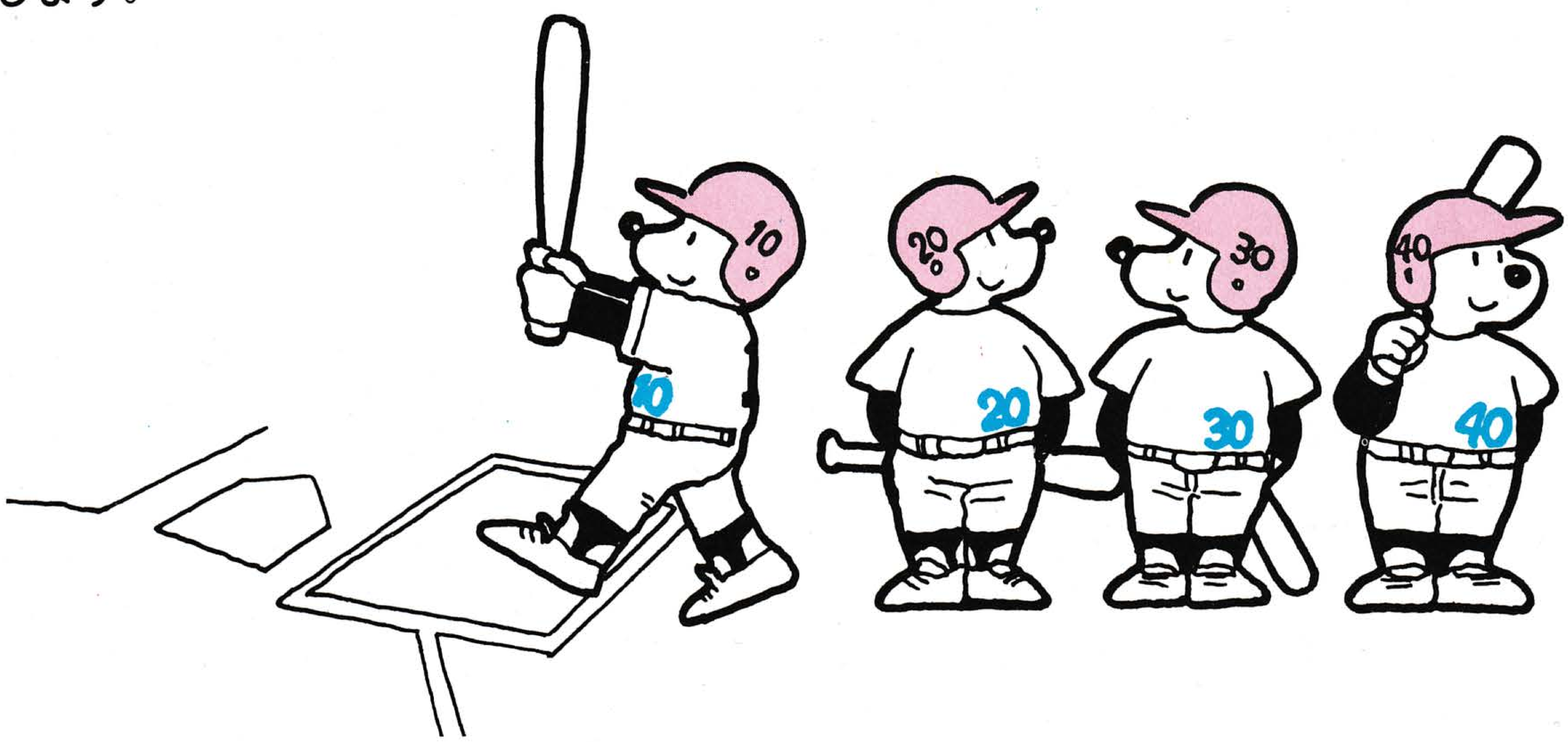
このような表示を**エラーメッセージ**と呼びます。

正しく命令を入力したつもりでいるときにいきなりエラーメッセージが表示されると、誰でもビックリして不愉快になります。
しかし、ＢＡＳＩＣでプログラムの作りかたを覚えようとする人にとって、エラーメッセージは大切な味方です。
エラーメッセージにはいくつもの種類があり、起きたエラー（まちがい）の内容に応じたメッセージ（知らせ）が表示されます。つまり、表示されたエラーメッセージの意味を調べることによって、何がいけなかったのかをある程度知ることができるのです。

表示されたエラーメッセージの意味は、195ページの「ＢＡＳＩＣエラーメッセージ表」を参照すれば分かります。
エラーメッセージが表示されてもイライラしたりせず、うまく利用してＢＡＳＩＣを覚えるために役立ててください。

プログラムに挑戦しよう

プログラムは、いくつかの命令をパソコンに分かる形式でまとめたものです。
ＢＡＳＩＣの命令のひとつひとつは小さな働きしかもっていません。しかし、たくさんの命令を組み合わせ、命令を実行する順番を指定し、パソコンが連続して働けるようにしてから「始め！」と号令をかけてやることによって、パソコンに大きな仕事をさせることができます。
つまり、プログラムといってもむずかしいことではなく、ひとつひとつの命令を実行させる順番に並べたものにすぎません。
命令を実行させる順番は、それぞれの命令を書いた行の先頭に付けた番号（行番号）で指定します。パソコンは、特別な命令を除いて、行番号が小さい行の命令から順に実行します。



それではここで、プログラムを作る作業の基本的なステップを説明しておきます。

① **パソコンにさせる仕事をハッキリ決める。**

② **その仕事をできるだけ細かな部分に分けて考える。**
次ページ下段のような図にして考えると便利です。

③ **それぞれの部分の仕事を、ＢＡＳＩＣの命令に置き換える。**
「資料編」の「ＢＡＳＩＣリファレンス表」を参照すると便利です。

④ **実行される順番に、命令に行番号を付けてプログラムにする。**
慣れないうちは、①～④の作業はむずかしいものです。初めのうちはこの本や他のＢＡＳＩＣの解説書などに載っているプログラムをパソコンに入力し、実行させながら理解するようにしましょう。

⑤ **できたプログラムをパソコンに入力する。**
プログラムの入力のしかたについては、28ページから説明しています。

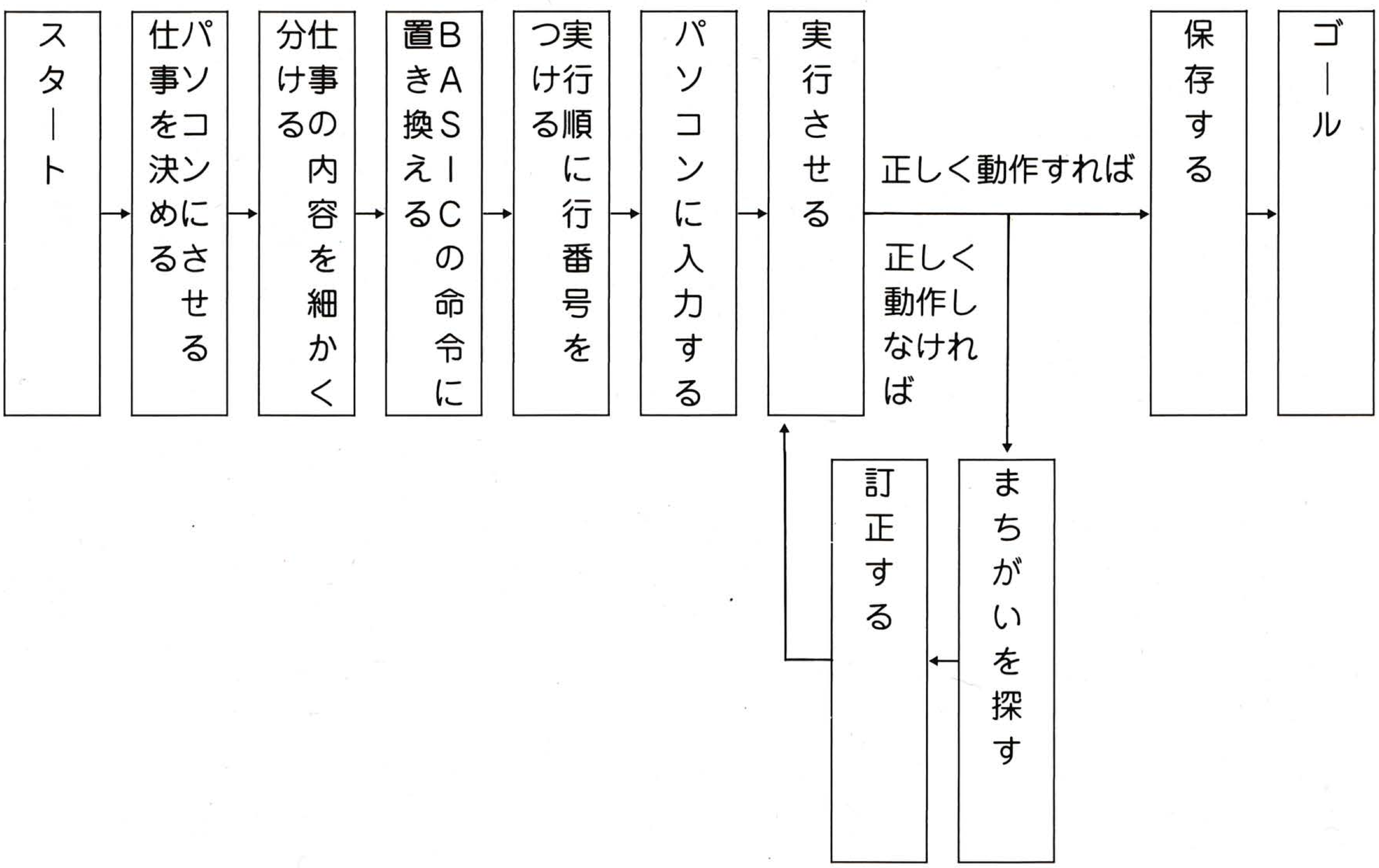
⑥ **プログラムを実行させる。エラーメッセージが表示されたり、正しく動作しなかったときには、プログラムをもう一度確かめ、まちがいを捜して修正する。**

プログラムの実行のさせかたと、エラーメッセージが表示されたときにどうしたら良いかは、30ページから説明しています。

⑦ **完成したプログラムをフロッピーディスクに保存する。**

プログラムの保存や呼び出しのしかたは、３４ページから説明しています。
プログラムの入力を途中で中断するようなときでも、それまで入力した部分を保存しておけば、あとで、保存した部分を呼び出して入力を続けることができます。

プログラムを作るときの仕事の流れ 例)



九九(くく)を表示するプログラム

それでは、かんたんなプログラムの例として、かけ算の九九を表示するプログラムを作ってみましょう。

パソコンにさせる仕事は、**「キーボードから入力された数を１～９までの数にかけて、そのたびに式と答えを画面に表示すること」**と決めます。

パソコンがする仕事の内容を細かく分けると、次のようになります。

① **もとになる数の入力を求める。**
② **入力された数に１をかける。**
③ **かけ算の式と答えを画面に表示する。**
④ **②、③の仕事を、かける数を２～９まで変えながら繰り返す。**

それぞれの仕事の内容をＢＡＳＩＣの命令に置き換えると、次のプログラム例のようになります。

行番号	命令	対応する仕事の番号
10	INPUT "(1−9)";K	①
20	FOR N=1 TO 9	④
30	L=K*N	②
40	PRINT K;"*";N;"=";L	③
50	NEXT N	④

上のプログラム例の各命令は、次のような働きをします。

```
10 INPUT "(1−9)";K
```
K：変数　──①の仕事に対応

キーボードから入力された数値を、Ｋという名前の入れ物（**変数**）の中に入れます。
（変数については、４４ページでくわしく説明しています）
上のプログラム例では、"　"に囲まれた文字「（１－９）」と「?」を表示し、数値の入力を待ちます。

""の中の文字と、変数の名前は、自由に変えることができます。（上のプログラムでは、変数の名前は30行・40行と同じにすることが必要です。）""とその中の文字は省略することができます。

```
20 FOR N=1 TO 9 ――― ④の仕事に対応
       変数
50 NEXT N
        同じ変数
```

変数の数値を1，2，3，‥‥9と1つずつ増しながら、FOR命令とNEXT命令の間にある各命令を繰り返し実行します。Nの値が9を越えると、繰り返しをやめて次の行に進みます。

```
30 L=K*N ――― ②の仕事に対応
```

K×Nを計算し、計算結果を変数Lに入れます。

例） かけ算のもとになる数に5を入力したときは、FOR，NEXT命令の繰り返しの1回目ではK＝5，N＝1ですので5×1＝5となり、変数Lには5が入ります。同様に2回目の繰り返しでは、5×2＝10になり、Lには10が入ります。

```
40 PRINT K;"*";N;"=";L ― ③の仕事に対応
```

計算結果をKの値、＊、Nの値、＝、Lの値の順に画面に表示します。表示する項目の間に「;」がある場合は、前の項目に続けて次の項目を表示します。表示が終わると、次に文字が表示される位置はすぐ下の行の左端になります。

例） K＝5，N＝1，L＝5の場合

```
5 * 1 = 5
```

メモ：●行番号は1、2、3…でもかまいませんが、あとで行を追加したりするときのために、10、20、30…と10間隔でつけるのがふつうです。

●FORとNEXTについては、48ページでくわしく説明しています。

それではこのプログラムを入力し、実行してみましょう。入力のしかた、実行のしかたは次のページから説明しています。

プログラムを入力しよう

新しくプログラムを入力する前に

プログラムを入力するときには、まず、ダイレクトモードで次の命令を入力して実行させます。

N E W ↵

NEW（ニュー）命令は、現在パソコンが覚えているプログラムをすべて忘れさせる命令です。
パソコンがあるプログラムを覚えている状態のままで別のプログラムを入力してしまうと、２つのプログラムがひとつに混ざり合ってしまいます。
いまの場合はパソコンが覚えているプログラムは無いはずですが、プログラムの入力の前には必ずNEW命令を実行させる習慣を付けてください。

プログラムの入力のしかた

- **必ず行番号を行の先頭につけ、それに続けて命令などを入力します。**
- キーを押しまちがえたときの修正のしかたは、下図を参照してください。
- １行が長いときには、入力を続けているあいだにカーソルが画面の右端から下の行の左端に移動することがありますが、かまわずに入力を続けます。
- **１行分の入力が終わるたびに、忘れずに ↵ を押します。**（ただし、画面で２行以上になるときは、途中の行で押してはいけません。）このとき、カーソルは必ずその行のどこかにあることが必要です。

修正のしかた		操作の手順
文字の上書き（まちがった文字を入力したとき）		CAS ⇨ CAS ［L］ ⇨ CLS
文字の削除（不要な文字を入力したとき）	BS	CALS ⇨ CALS ［BS］ ⇨ CLS（カーソルの前の文字を消す）
	DEL	CALS ⇨ CALS ［DEL］ ⇨ CLS（カーソルのある位置の文字を消す）
文字の挿入 INS（文字を入力し忘れたとき）		CS ⇨ CS ［INS］ ⇨ CS ［L］ ⇨ CLS ［INS］ ⇨ CLS

たとえば、このプログラムの10行を入力するときには、次のようにキーを押すことになります。
（下の［ ］は、［SPACE］をあらわします。）

［1］［0］［ ］［I］［N］［P］［U］［T］［ ］［"］
［(］［1］［-］［9］［)］［"］［;］［K］［↵］

26ページのプログラムを、上のやりかたで入力してください。

行番号を付けて命令を入力すると、↵ を押しても命令は実行されません。
つまり、行番号を付けた命令を入力されると、パソコンは自動的にプログラムだと判断するわけです。

まちがいやすい文字に気をつけよう!

プログラムを入力するとき、よく起こすのはキーの打ちまちがいとリストの読みまちがいです。
次の文字は、特にまちがいやすいので注意してください。

- **「O（オー）」と「0（ゼロ）」**
- **「I（アイ）」と「l（小文字のエル）」と「1（いち）」**
- **「.（ピリオド）」と「,（カンマ）」**
- **「:（コロン）」と「;（セミコロン）」**
- **「S（エス）」と「5（ご）」**
- **「ー（ハイフン、マイナス）」と「ー（カタカナの長音）」**

他には、文字の抜け（特に「"」「（ ）」など）や、つづりのまちがいに気をつけてください。

プログラムの実行ー失敗しても当たり前

それでは、「かけ算の九九を表示するプログラム」を実行させてみましょう。

プログラムを実行する(RUN)

RUN（ラン）は、プログラムを実行させる命令です。　（ F5 を押しても同じです。）

プログラムの入力が終わったら、次のように入力してください。

R U N ↵

画面に「（1－9）？■」と表示されたら、1～9までのどれかの数字キーを押してから ↵ キーを押します。

```
RUN
(1-9)? 3
 3 * 1 = 3
 3 * 2 = 6
 3 * 3 = 9
 3 * 4 = 12
 3 * 5 = 15
 3 * 6 = 18
 3 * 7 = 21
 3 * 8 = 24
 3 * 9 = 27
Ok
■
```

例として、3を入力してみます。
プログラムが正しく動いていれば、画面に左のように表示されます。

もし、画面に何も表示されなかったり、何かエラーメッセージが表示されたら、プログラムを入力したときにどこかをまちがえています。
でも、ガッカリする必要はありません。次のページからプログラムの修正のしかたを説明しますので、あきらめずに正しく動くまで訂正してください。

プログラムを訂正する

入力したプログラムを画面に表示させてまちがいを探し、訂正します。

画面や紙の上に表示されたプログラムを**「リスト」**と呼びます。
リストの全体を表示させる場合は、次のように入力します。

L I S T ↵

LIST（リスト）は、入力したプログラムのリストを画面に表示させる命令です。
リストが長すぎてひとつの画面に表示し切れないときは、最初の行から順に画面の外に消えていきます。これを**「スクロールする」**と呼びます。

・**スクロールを一時停めたいときは、**［STOP］を押します。
このときは、リストを見てまちがいを探すだけで、プログラムの訂正はできません。再び［STOP］を押すと、スクロールを再開します。

・**スクロールを中止させたいときは、**［CTRL］を押しながら［STOP］を押します。

その他、

・**特定の１行だけを表示させたいとき** …… LIST 20↵
上の例では、２０行だけを表示します。

・**ある範囲の行を表示させたいとき** ……… LIST 20−40↵
２０行〜４０行のすべての行を表示します。

・**先頭から途中の行までを表示させたいとき** LIST −30↵
先頭の行から３０行までを表示します。

・**途中から最後の行までを表示させたいとき** LIST 30−↵
３０行からあとのすべての行を表示します。

［−］（マイナス）と［ー］（カタカナの長音）をまちがえないよう注意してください。ＢＡＳＩＣの命令では、**マイナス**だけを使用します。

エラーメッセージが表示されたときは

まちがいの内容によっては、プログラムの実行が中断され、画面にエラーメッセージが表示されてダイレクトモードに戻ることがあります。
この状態を**「エラーになる」**といいます。
プログラムを実行中にエラーになったときは、エラーメッセージの後ろにエラーになった行の行番号が表示されます。

例) Syntax error in 30

この場合、表示された番号の行にまちがいがあるときと、他の行のまちがいが原因になってその行でエラーになったときの二通りがあるので注意してください。

表示されるエラーメッセージは、エラーの種類を大まかにしめしています。それぞれのエラーメッセージの意味は、195ページの「ＢＡＳＩＣエラーメッセージ表」にのっています。まちがいを探す参考にしてください。

まちがいを見つけたら

まちがいが見つかったら、画面のリストを正しく書き換え、↵ を押します。

- ↵ は、1行書き換えるたびに押してください。
- ↵ を押すときには、カーソルを書き換えた行に合わせておいてください。他の行にカーソルを動かしてから ↵ を押しても、プログラムは修正されません。
- 書き換えているあいだにカーソルが次の行まで届いてしまったときは、↵ を押す前に、DEL を使って次の行を画面から完全に消してしまってください。カーソルが次の行に届いたままで ↵ を押すと、2つの行がくっついてしまいます。

訂正が終わったら、ＬＩＳＴ命令を使って確かにプログラムが訂正されたかどうか確かめます。書き換えたとおりになっていたら、ＲＵＮ命令を入力してもう一度プログラムを実行させてみます。まだ正しく動かないようなら、他にもまちがいがありますから、もう一度まちがいを探して訂正してください。

《ちょっと一言》

大きなプログラムの修正に便利な命令

DELETE　（デリート）……指定された行を消去します。

例）

DELETE 50	（ 50行を消去）
DELETE 100−200	（100行～200行のすべての行を消去）

RENUM　（リナンバー）……行番号を付け替えます。

例）

RENUM	（10行から10間隔で行番号を付け直す）
RENUM 100, 30	（30行からあとのすべての行を、100行から10間隔の行番号に付け直す）

プログラムの保存と呼び出し

パソコンは、電源スイッチが切れた瞬間に、入力されたプログラムを忘れてしまいます。そこで、入力したプログラムを保存しておく方法を説明します。
一度プログラムを保存しておけば、いつ電源スイッチを切ってもだいじょうぶ。
いつでも、保存しておいたプログラムをパソコンの中に呼び出すことができるからです。

ディスクへの保存と呼び出しに必要なもの

・フロッピーディスクドライブ（ただし、ドライブ内蔵のパソコンでは必要ありません）
・フロッピーディスク（3.5インチ２ＤＤまたは3.5インチ１ＤＤと呼ばれるもの）
（※接続するフロッピーディスクドライブの機種によっては、接続用のカートリッジ（フロッピーディスクコントローラ）を別に買い求めることが必要です。）

どこで手に入れる?

フロッピーディスクドライブやディスクは、パソコンを専門に売っているお店（パソコン専門店）で売られています。また一般の電器店でも、パソコンを売っているお店なら手に入ることがあります。ただし、これらの機器を買い求めるときには、次のことをお店の人に確かめてください。

フロッピーディスクドライブについては、
・**「ＭＳＸ用」か？**
　ＭＳＸ用以外のフロッピーディスクドライブは接続できません。
・フロッピーディスク**コントローラが内蔵**されているか？
　もし内蔵されていない機種なら、コントローラを別に買い求めることが必要です。

ディスクについては、
・**「3.5インチ２ＤＤ」**（両面）または**「3.5インチ１ＤＤ」**（片面）と呼ばれる規格のものか？
　それ以外の規格のディスクを使うと、プログラムの保存や呼び出しができません。

実際の操作は?

以降のページでは、次のことを説明します。

次のことは、機種による違いがあるので、この本では説明しません。
これらについては、あらかじめ、パソコンとフロッピーディスクドライブの説明書を読んでおいてください。

・フロッピーディスクドライブとパソコンの接続のしかた
・フロッピーディスクドライブを接続したときの電源スイッチの入れかた
・ディスクの出し入れのしかた

新しいディスクを使う前に必要な準備

お店で買って来たばかりの何も保存されていない新しいディスクを、この本では、
「生（なま）ディスク」と呼ぶことにします。
（パソコンの付属ソフトのフロッピーディスクや、ゲームやワープロなどのソフトが入ったディスクは、生ディスクではありません。）

生ディスクは、そのままでは使えません。まだ、**「フォーマット」**（または初期化）と呼ばれる準備をしていないからです。

フロッピーディスクの中には、**「磁気シート」**と呼ばれるごく薄い円盤状のフィルムが入っています。保存したプログラムやデータは、すべてこのシートに磁気の強弱の形で書き込まれます。
生ディスクの磁気シートは白紙のような状態で、何も目印になるものがありません。そこで生ディスクを使う前には、ちょうどノートのケイ線や原稿用紙のマス目のように、プログラムなどを書き込む位置を決める目印を書き込むことが必要です。これが「フォーマット」です。

> **ご注意**：すでに何かが保存されているフロッピーディスクをフォーマットすると、プログラムやデータが消えてしまいますから注意してください。

フォーマットのしかたは、次ページから説明しています。

ディスクをフォーマットする(CALL FORMAT)

以下の説明は、次のどちらかの場合について説明しています。
・内蔵ドライブ付きのパソコンでドライブを増設していない場合
・内蔵ドライブ無しのパソコンでドライブを1台だけ接続している場合
また、すでにＢＡＳＩＣがスタートしているものとします。

1

CALL FORMAT ↵ と入力します。

CALL FORMAT（コール・フォーマット）は、ディスクをフォーマットする命令です。

2

画面に下のように表示され、ドライブの指定を求めてきます。

```
Drive name? (A, B) ■
```

画面の表示は、パソコンと、接続しているフロッピーディスクドライブの機種およびドライブの合計数によって違います。

3

[A] を入力します。(ドライブAを指定)

画面に次のように表示され、フォーマットの種類を聞いてきます。

```
1 — 1 side, double track
2 — 2 sides, double track

? ■
```

画面の表示は、パソコンと接続しているフロッピーディスクドライブの機種によって違います。

4

[2] を押します。（3.5インチ２ＤＤディスクのとき。１ＤＤなら [1] を押す）

画面に次のように表示され、用意ができているかどうかの確認を求めてきます。

```
Strike a key when ready ■
```

5 ディスクのライトプロテクトタブが書き込み可能の位置にあることを確かめてから、フロッピーディスクドライブ（ドライブを増設しているときは、増設したドライブ）に、生ディスクを差し込み、⏎ を押します。
（文字キーや［SPACE］を押してもかまいません。）

6 フロッピーディスクドライブのアクセスランプが点滅して、フォーマットを始めます。
※フォーマットには、少し時間がかかります。必要な時間は機種によって違いますが、一般的には 2～3 分で終わります。

7 フォーマットが終わると画面に右のように表示され、アクセスランプが消えます。

```
Format complete
Ok
■
```

8 ディスク取り出しボタンを押し、フォーマットしたディスクを取り出します。

保存するプログラムの名前を決める（ファイルスペック）

プログラムを保存するときには、まずプログラムに名前を付けます。プログラムを呼び出すときには、その名前によって他のプログラムと区別します。この名前を**「ファイルスペック」**と呼びます。ファイルスペックは次ページのＳＡＶＥ命令やＬＯＡＤ命令などといっしょに入力します。

例）

"A:MYPROG1.BAS"

- デバイス名：A:
- ファイル名：MYPROG1
- ピリオド：.
- 拡張子：BAS

デバイス名の部分には、保存に使うフロッピーディスクドライブの**ドライブ名**と、**「：」（コロン）**を書きます。

・ドライブ名は、A～Hの英文字一字で指定します。
・ドライブが１台だけの場合は、ドライブ名はAになります。
・使用するドライブがドライブAのときは、デバイス名を省略できます。
・ドライブが２台以上ある場合は、接続の方法と機種によって、ドライブ名が変わります。
　一般的には、ドライブ１台内蔵のパソコンにドライブを１台増設したときには、増設したドライブがA、内蔵ドライブがCになります。

ファイル名は、8文字以内の英文字、数字、カタカナで自由に書きます。
・英字の大文字と小文字は区別されず、すべて大文字とみなされます。

拡張子の部分は、「.」（ピリオド）の後ろに3文字以内で自由に書きます。
（カタカナや記号、小文字については、ファイル名と同じです。）
・拡張子の使いみちは自由です。プログラムの分類や新旧を区別するのに便利です。
・拡張子は省略できます。ただし、拡張子を付けて保存したプログラムを呼び出すときは省略できません。

では、「かけ算の九九を表示するプログラム」を、前ページの例のファイルスペックでディスクに保存してみましょう。

プログラムをディスクに保存する(SAVE)

以下の説明では、すでにプログラムがパソコンに入力されているものとします。
プログラムを保存することを**「セーブ」**と呼びます。

1. フォーマット済みのディスクを、ライトプロテクトタブが書き込み可能の位置にあることを確かめてからドライブAに入れます。

2. 次のように命令とファイルスペックを入力します。

   SAVE "A:MYPROG1.BAS" ↵

   SAVE（セーブ）は、プログラムをディスクに保存する命令です。

3. ドライブAのアクセスランプが点灯し、セーブが始まります。
   セーブが終わると、画面に「Ok」と表示され、アクセスランプが消えます。

4. アクセスランプが消えていることを確かめ、ディスク取り出しボタンを押して保存したディスクを取り出します。

プログラムをディスクから呼び出す(LOAD)

保存したプログラムをパソコンのメモリの中に呼び出すことを**「ロード」**と呼びます。
ＢＡＳＩＣのプログラムをディスクからロードするときには、次のようにします。
（以下の説明では、すでにプログラムをディスクに保存しているものとします。）

1

プログラムを保存したディスクをＡドライブに入れ、次のように入力します。

```
LOAD "A:MYPROG1.BAS" ↵
```

ＬＯＡＤ（ロード）は、ディスクに保存したプログラムをパソコンの中に呼び出す命令です。必ず保存したときと同じファイル名と拡張子を入力してください。デバイス名は、使用するドライブのドライブ名に変えることができます。

2

ドライブＡのアクセスランプが点灯し、ロードが始まります。
ロードが終わると「Ｏｋ」と表示され、アクセスランプが消えます。

3

ディスクを取り出すときは、アクセスランプが消えたことを確かめてからディスク取り出しボタンを押し、ディスクを取り出します。

これで、プログラムはパソコンの中に呼び出されています。
ＬＩＳＴ命令を使って確かめてください。

ファイルスペックを確かめる(FILES)

プログラムを保存してから時がたつと、保存したプログラムのファイルスペックを忘れてしまうことがあります。
このとき、まちがえて同じディスクに保存している他のプログラムのファイルスペックを使って保存したりすると、保存されていたプログラムがそのプログラムに書き換えられてしまいます。
プログラムを保存したり呼び出したりする前には、なるべくそのディスクに保存されているすべてのプログラムのファイルスペックを確かめるようにしましょう。

ＦＩＬＥＳ（ファイルズ）は、ディスクに保存されているすべてのプログラムやデータのファイルスペックを画面に表示させる命令です。

ＦＩＬＥＳ命令の基本の形は、次のようになります。

```
FILES "ドライブ名:"
```

ドライブ名を省略すると、ドライブＡに入っているディスクを調べ、ファイルスペックの一覧を画面に表示します。リストの表示と同様、一度に画面に表示しきれないときは初めに表示されたファイルスペックから順に消えて行きます。
（STOP を押すと画面の表示を一時止めることができ再び押すと再開します。）

ディスクの入っていないドライブを指定すると、下のエラーメッセージが表示されます。

```
Disk offline
```

これで、ディスクへの保存と呼び出しに必要な命令の説明は終わりです。この他にも、ＢＡＳＩＣにはディスクに保存したプログラムを管理するための便利な命令がありますので、ついでにここで紹介だけしておきます。

・ＣＯＰＹ（コピー）　……… ディスクに保存したプログラムを、別のディスクにコピーします。（これを「バックアップをとる」と呼びます）
　例）　COPY "A:PROG1.BAS" TO "B: "
※例のように入力すると、アクセスランプが点灯したあと、画面に下の表示が出ます。

```
Insert diskette for drive B: ←（またはA:）
and strike a key when ready ■
```

この表示が出るたびにドライブの中のディスクを入れ替え、⏎を押してください。コピーが終わると、「Ｏｋ」と表示されます。(入力したファイルスペックのファイルがコピーされます。)

・ＫＩＬＬ（キル）　………… ディスクに保存したプログラムを消去します。
　例）　KILL "NOUSE.BAS"

・ＬＦＩＬＥＳ（エルファイルズ）ディスクに保存されているすべてのプログラムやデータのファイルスペックを、プリンタで印刷します。（プリンタが必要です）
　例）　LFILES

・ＮＡＭＥ（ネーム）　……… ディスクに保存したプログラムの、ファイルスペックを付け替えます。
　例）　NAME "PROG1.BAS" AS "PROG2.BAS"
　　（"PROG1.BAS"＝旧ファイル名、"PROG2.BAS"＝新ファイル名）

・ＲＵＮ　"　"（ラン）　…． ＲＵＮ命令に続けてフロッピーディスクに保存されているプログラムのファイルスペックを入力すると、そのプログラムを呼び出したあと、自動的に実行します。
　例）　RUN "MYPROG1.BAS"

ＢＡＳＩＣは楽しみながら覚えるモノ。とにかく入力して動かしてみましょう。

体験！プログラミング



BASICの約束ごと

新しい命令を説明する前に、ぜひ知っておいて欲しいことをかんたんにまとめて説明します。
これらは、ＢＡＳＩＣでプログラムを作るときに必ず守らねばならない約束ごとです。

変　数

変数とは、プログラムの中などで使う数値や文字列を覚えておくための**「入れ物」**のことです。
（26ページのＫ，Ｎ，Ｌがこれに当たります）

・変数名

変数には、２文字までの英文字または英文字と数字で名前を付けますが、次のようなルールにしたがってください。

① 先頭の一文字は英文字であること。
② 英文字の大文字と小文字は区別しない。
　「Ａ１」と「ａ１」は同じ変数とみなされる。
③ 区別するのは先頭から２文字だけで、３文字目からは無視される。
　「ＡＢ」、「ＡＢＣ」、「ＡＢ１」は同じ変数とみなされる。
　ただし、型宣言文字（⇨文字型変数）は別。
④ ＢＡＳＩＣの予約語（⇨188ページ）を含む名前は使用できない。

・文字型変数

文字列を覚えるための**文字型変数**は、数値を覚えるための**数値型変数**と区別するため、変数名の後ろに「$」（ダラー）を付けます。「$」のように、変数の型を区別するための文字を**「型宣言文字」**と呼びます。

例）　「Ａ$」、「Ｂ１$」、「ＡＢ$」

・変数に数値や文字列を覚えさせるには

数値型変数　　変数名＝数値または演算式

例）	Ａ１＝３５００	変数Ａ１に 3,500を入れる。
	ＰＴ＝Ａ１＋３００	変数ＰＴに（Ａ１の値）＋300の計算結果を入れる。
	Ｙ＝Ｙ＋２	変数Ｙに（現在のＹの値）＋２の計算結果を入れる。

文字型変数　　変数名＝文字列または文字列の式

例）	ＭＪ$＝″ＭＯＪＩ″	変数ＭＪ$に文字列ＭＯＪＩを入れる。
	Ｐ$＝Ｋ$＋″かい″	変数Ｐ$に（Ｋ$の値の文字列）と「かい」をつないだ文字列を入れる。

BASICで扱える数値の限界

有効桁数　：14桁まで。14桁を越える場合は、15桁目で四捨五入。

例）　123.456789012345 ──→ 123.45678901235

数値の範囲：10^{-64} ～10^{62}

極端に大きい数字や小さい数字は、指数表示になります。

例）

数式の表示	ＢＡＳＩＣの表示
1.23×10^{-30}	1.23Ｅ－30
456×10^{36}	4.56Ｅ＋38

画面の1行とプログラムの1行

プログラムでいう**「１行」**とは、ひとつの行番号に続いて入力されたすべての文字をさします。プログラムの１行には、行番号や空白などを含め、最大 254文字まで入力できます。したがってプログラムの１行は、画面上の１行におさまらず、２つ以上の行にまたがって表示される場合があります。

プログラムの１行の区切りが分かりにくくなったときは、ＬＩＳＴ命令でプログラムのリストを表示し、調べたいプログラム行にカーソルを合わせて ↵ を押してください。このとき、カーソルが表示された位置が次のプログラム行の先頭です。

文と複文

１つの命令を実際に入力するときには、命令だけでなく、いっしょに変数名や演算子などを入力することがよくあります。プログラムでは、これらを含めて**「文」**と呼びます。（文は命令だけでもかまいません）プログラムの１行には、２つ以上の文を**「：」**（コロン）で区切って入力することができます。このような行を**「複文」**または**「マルチステートメント」**と呼びます。

例）

```
10 CLS
20 INPUT "にゅうりょく";A :PRINT A
30 PRINT "このフ゜ロク゛ラムは、いみか゛ないフ゜
ロク゛ラムて゛す。"
```

（CLS ── 文、20行 ── 複文、30行 ── プログラムの１行）

白いヨーヨー

プログラムの入力や実行、それに保存と呼び出しのしかたが分かったら、いろいろなプログラムを入力して動かしたり、一部を変えてみるのが命令を覚える早道です。ここからは短いプログラムの例をあげ、出てくる新しい命令の使いかたをかんたんに説明します。



テキスト画面で白いヨーヨーを動かします。ヨーヨーが上下に動くたびに、音（ビープ音）が鳴ります。プログラムを止めるときは、CTRL を押しながら STOP を押します。
ヨーヨーとひもの表示には、グラフィック文字「●」と「|」を使います。グラフィック文字は GRAPH を押しながら文字キーを押して入力します。
表示するタテの位置は、繰り返しの回数を決める変数Yの値で指定しています。110行〜150行の繰り返しでヨーヨーを下げ、160行〜200行で上げています。

リストと解説

リスト	解説
`1 'save"ヨーヨー.bas"`	プログラムを保存するときのファイルスペック
`100 SCREEN 1:WIDTH 29:CLS`	画面の初期化と消去
`110 FOR Y=0 TO 20`	Yの値が20になるまで120〜140行を繰り返す
`120   LOCATE 14,Y  :PRINT "|";`	ヨーヨーのひもを表示する
`130   LOCATE 14,Y+1:PRINT "●";`	ヨーヨーを表示する
`140 BEEP`	音を鳴らす
`150 NEXT Y`	Yの値を１増して110行に戻る
`160 FOR Y=20 TO 0 STEP -1`	Yの値が０になるまで170〜190行を繰り返す
`170   LOCATE 14,Y  :PRINT "●";`	ヨーヨーを表示する
`180 LOCATE 14,Y+1:PRINT " ";`	前に表示したヨーヨーを消す
`190 BEEP`	音を鳴らす
`200 NEXT Y`	Yの値を１減らして160行に戻る
`210 GOTO 110`	110行にジャンプする

新しい命令の説明

REM（リマーク）

REMまたは「'」（アポストロフィ）は、同じ行の後ろの文字を無視させる命令です。（1行）

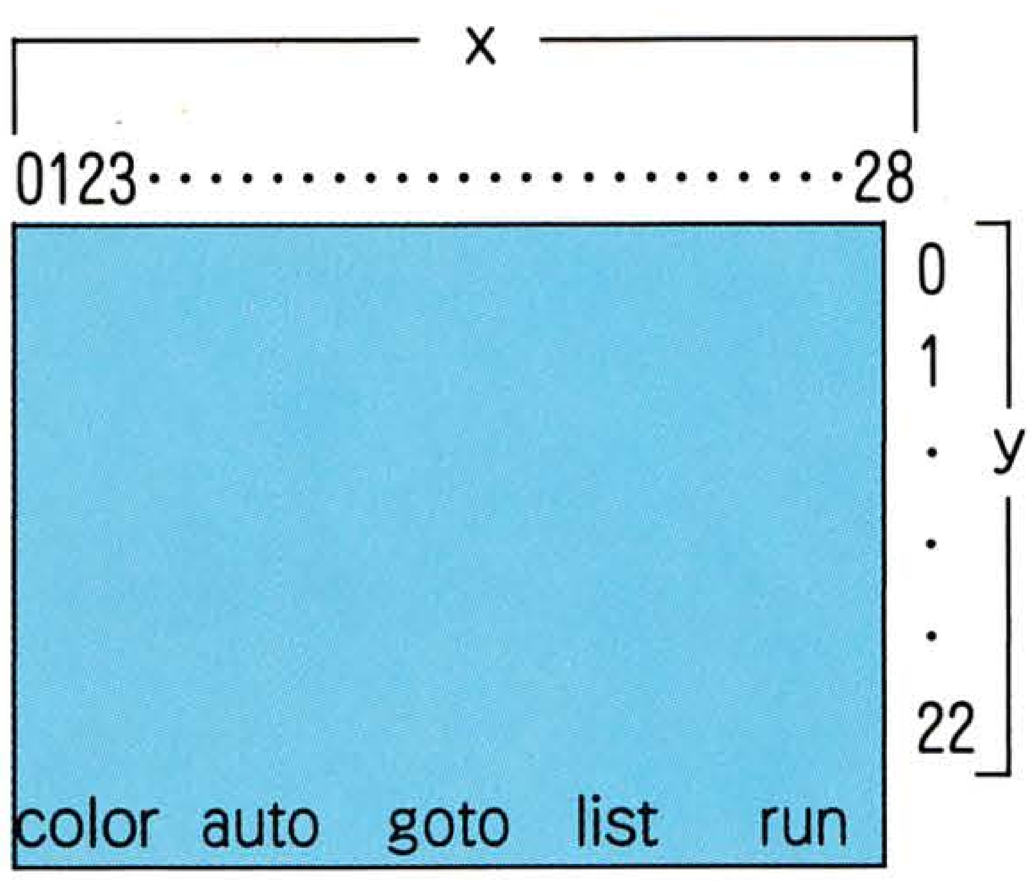
プログラムの中にちょっとした説明を入れるときなどに使います。必ず入力する必要はありません。ただし、REM文を省略するときは、REMや「'」だけでなくその行全体を省略します。
このプログラムの1行では、プログラムをセーブするときのファイルスペックを提案しています。

LOCATE（ロケート）

PRINT文などで文字を表示する位置を指定する命令です。

```
120 LOCATE 14, Y
```

14：ヨコ座標　Y：タテ座標



・ヨコ、タテの位置を1文字単位の数値で指定します。
・数値は0から始まり、画面の左上角が「0，0」にあたります。
・指定の順序は、常にヨコ座標を先に指定します。
・ヨコ座標、タテ座標とも、変数を使って指定できます。

《テクニック》

・SCREEN文やWIDTH文を実行すると、画面の右端の値（28）が変わります。
・KEY OFF命令を実行すると、画面の最下行も使用できます。（Y：0〜23の範囲）

PRINT（プリント）

画面に文字や数値を表示する命令です。

```
130 PRINT "●" ;
```

"●"：文字または数値　;：区切り記号

・PRINT文の最後に「;」を付けて実行させると、次にPRINT文を実行したとき、その後ろから続けて文字や数値が表示されます。
（ただし、LOCATE文で表示位置を指定し直したときを除きます）
・PRINT文の最後に「;」や「，」を付けないで実行させると、次にPRINT文を実行させたとき、その下の行の左端から表示されます。
・「;」や「，」を**「区切り記号」**と呼びます。区切り記号を使うことにより、一つのPRINT命令で2つ以上の表示項目を表示することができます。
・表示項目の間を「;」で区切ると、数値や文字列を続けて表示します。（ただし、数値はそれぞれ前後にひと桁分の間隔が開けられます）
・表示項目の間を「，」で区切ると、数値や文字列の間を離して表示します。
（後ろの項目の表示位置は、SCREEN文やWIDTH文によって変わります）

FOR, NEXT（フォー、ネクスト）

ＦＯＲ文とＮＥＸＴ文の間にある各文を、指定された回数だけ繰り返し実行する命令です。

```
110  FOR  Y=0    TO  20
          変数名 開始値    終了値
 :
150  NEXT  Y
           同じ変数名    ※変数名は数値型変数名に限る。
```

・ＦＯＲ文とＮＥＸＴ文は、必ず組み合わせて使います。対応するＦＯＲ文を実行しないうちにＮＥＸＴ文を実行させるとエラーになります。
・ＦＯＲ文とＮＥＸＴ文の変数名は、同じであることが必要です。
・ＦＯＲ文を実行すると、変数の値を開始値にして次の文を実行します。
・ＮＥＸＴ文を実行するたびに、変数の値に１を加えてから終了値と比較し、開始値～終了値の範囲を越えていなければＦＯＲ文の次の文から繰り返します。
・変数の値が開始値～終了値の範囲を越えると、繰り返しを終えてＮＥＸＴ文の次の文に進みます。

```
160 FOR Y=20 TO 0 STEP -1
          開始値  終了値   増分値
```

・ＦＯＲ文でＳＴＥＰ（ステップ）と増分値を入力したときは、ＮＥＸＴ文を実行したときに１を加える代わりに増分値を加えます。
・増分値は、負（マイナス）の値でもかまいません。このときは、開始値を終了値より大きくします。
・開始値、終了値、増分値は、変数でも指定できます。
・ＮＥＸＴ文の変数名は省略できます。省略したときは、最後に実行したまだ終了していない繰り返しのＦＯＲ文の、次の文から繰り返します。

GOTO（ゴートゥー）

行番号で指定された行にジャンプし、その行の先頭の文から順に実行する命令です。

```
210  GOTO  110
           行番号
```

・行番号は、プログラム中に実在する行であることが必要です。省略はできません。
また、変数は使えません。

プログラムの変更

もっとたくさんのヨーヨーが表示されるよう、プログラムを変更しましょう。

プログラムを止めて、次のように入力してください。

```
115 FOR N=3 TO 23 STEP 5↵
135 NEXT N↵
165 FOR N=23 TO 3 STEP -5↵
185 NEXT N↵
```

現在プログラム中にない行番号で新たに入力した行は、プログラムに追加されます。

こんどは、次のように入力してください。

```
120 LOCATE N,Y:PRINT "|";↵
130 LOCATE N,Y+1:PRINT "●";↵
170 LOCATE N,Y:PRINT "●";↵
180 LOCATE N,Y+1:PRINT " ";↵
```

現在プログラム中にある行番号で入力すると、同じ行番号の現在の行が新しく入力した内容に書き換えられます。

入力が終わったら、ＬＩＳＴ命令でプログラムを確認してください。
正しく訂正できていたら、訂正前のプログラムとは異なる名前でセーブしておきましょう。

《ご注意》

・先にセーブしたプログラムと同じファイルスペックで同じディスクにプログラムをセーブすると、先にセーブしたプログラムが書き換えられて消えてしまいます。

数当てゲーム

パソコンが勝手に決めた数字を推理し、キーボードから答えを入力するゲームです。答えと正解が右下の図でタテまたはヨコの同じ列にあるときは、「おしい！」と表示されます。当たるまで答え続けて、何回で当てられるかを競ってください。



0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

途中で実行を中止したいときは、CTRL を押しながら STOP を押してください。

リストと解説

`1 'save"カズ゛アテ.bas"`	保存するときのファイルスペック
`100 SCREEN 1:WIDTH 29:CLS`	画面の初期化と消去
`110 R=INT(RND(-TIME)*25)`	乱数を使って正解を決める
`120 X=R MOD 5:Y=R ¥ 5:N=0`	正解がある位置の行と列を計算する
`130 '`	
`140 INPUT"よそう すうち (0-24) ";K:PRINT`	答えの入力を求め、変数Kに入れる
`150 IF K<0 OR K>24 THEN 130`	Kの値が範囲外なら、再び入力を求める
`160 N=N+1:X1=K MOD 5:Y1=K¥5`	答えの回数と答えの位置を計算する
`170 IF X=X1 AND Y=Y1 THEN200`	正解と答えの位置が同じなら200行へ
`180 IF X=X1 OR  Y=Y1 THEN A$="おしい!" ELSE A$="かんけいないもーん!"`	同じでないなら結果でＡ$の値を決める
`190 PRINT A$:PRINT:GOTO 130`	Ａ$の値を画面に表示し､130行に戻る
`200 '`	
`210 PRINT "おめでとう";N;"かいで アタリ !!"`	答えが正しいときの表示
`220 END`	プログラムの実行を終了する

実行させると、推理した答えの入力を求めてきます。０～24の数字を入力してください。答えが合っていれば、何回目で当たったかを表示してプログラムの実行を終了します。

新しい命令の説明

関数

関数は、あたえられた材料（数値または文字列）を使って一定の仕事をし、その結果を返す小さな工場のような働きをするものです。

たとえば、ＩＮＴという関数は、「２．５」という数値を材料としてあたえられると、「２」という整数値を返します。

この材料としてあたえた数値や文字列のことを**「引数」**（ひきすう）と呼びます。

関数は、引数として数値をあつかう**「数値型関数」**と文字をあつかう**「文字型関数」**の２種類に大きく分かれます。

《ご注意》

関数は、命令とは異なり、それだけで文を作ることができません。
必ず命令か「＝」と組み合わせて文を構成します。

INT（インテジャー）（関数）

ＩＮＴは、あたえられた数値を整数にして返す関数です。

例）

```
A＝ＩＮＴ（B）
数値型変数名　引数
```

・あたえられた引数の小数部分を切り捨て、整数部分を返します。
・引数は数値に限られます。定数（決まった数）でも変数でもかまいません。

RND（ランダム）（関数）

ＲＮＤは、数値をあたえられるたびに０以上１未満のでたらめな数値（乱数）を返す関数です。

例）

```
A＝ＲＮＤ（B）
数値型変数名　引数
```

・引数が正（プラス）の数値のとき……プログラムをスタートさせてからＲＮＤ文を実行した回数によって返す乱数の値が決まる
・引数が０のとき……………………前回に返した乱数と同じ値を返す
・引数が負（マイナス）の値のとき……引数の値によって決まった乱数を返す

TIME
(タイム)
(システム関数)

システム変数とは、ＢＡＳＩＣが用意する特別な変数のことです。
ＴＩＭＥは、1/60秒ごとに１ずつ値が増すシステム変数です。

値の範囲は０～65535までで、65535の次は０に戻ります。プログラムやダイレクトモードで値を入れることもできますが、範囲を越えた値を入れるとエラーになります。

例）

```
TIME=0
Ok
PRINT TIME
 297
Ok
■
```

297 ― この値は、ＴＩＭＥ＝０を入力してからＰＲＩＮＴ　ＴＩＭＥを入力するまでの時間によって変わります。

このプログラムでは、ＲＮＤ関数の引数としてＴＩＭＥを負の値に変えたものが使われています。

MOD, ¥
(モジュロ、エン)
(演算子)

ＭＯＤは、割り算の「余り」を求める演算子です。
¥は、割り算の「商」を求める演算子です。

例）

```
A=5 MOD 2:B=5 ¥ 2
Ok
PRINT A;B
 1  2
Ok
■
```

・ＭＯＤ，¥で扱える数値は、－32768～＋32767までです。

このプログラムでは、乱数で決めた正解や入力された答えが図の中のどの行と列に当てはまるかの計算に使われています。

=
(イコール)
(関係演算子)

これまでの説明では、＝は変数にある値を入れるときに使いましたが、このプログラムの170行と180行では、＝を違う役目で使っています。これらの＝は**「関係演算子」**です。

関係演算子は、演算子の左右の値の関係を判定する記号です。
＝は左右の値が等しいかどうかを判定します。

AND, OR
(アンド、オア)
(論理演算子)

論理演算子は、数値またはその他のデータを一定の条件にしたがって操作する論理演算で、条件を設定する演算子です。

・ＡＮＤは、日本語の「かつ」にあたります。「Ａ　ＡＮＤ　Ｂ」は「ＡであるとＢである」という意味になります。
・ＯＲは、日本語の「または」にあたります。「Ａ　ＯＲ　Ｂ」は「ＡであるかまたはＢである」という意味になります。

このプログラムでは、２つの関係演算の結果が両方とも成立している場合の判定にＡＮＤを、どちらか片方だけが成立している場合の判定にＯＲを使っています。

IF~
THEN~
ELSE~
(イフ・ゼン
・エルス)

ＩＦ～ＴＨＥＮ～ＥＬＳＥ～（略してＩＦ文）は、ＩＦとＴＨＥＮのあいだに書かれた条件が成立するかどうかによって、プログラムの流れを変える命令です。

例）

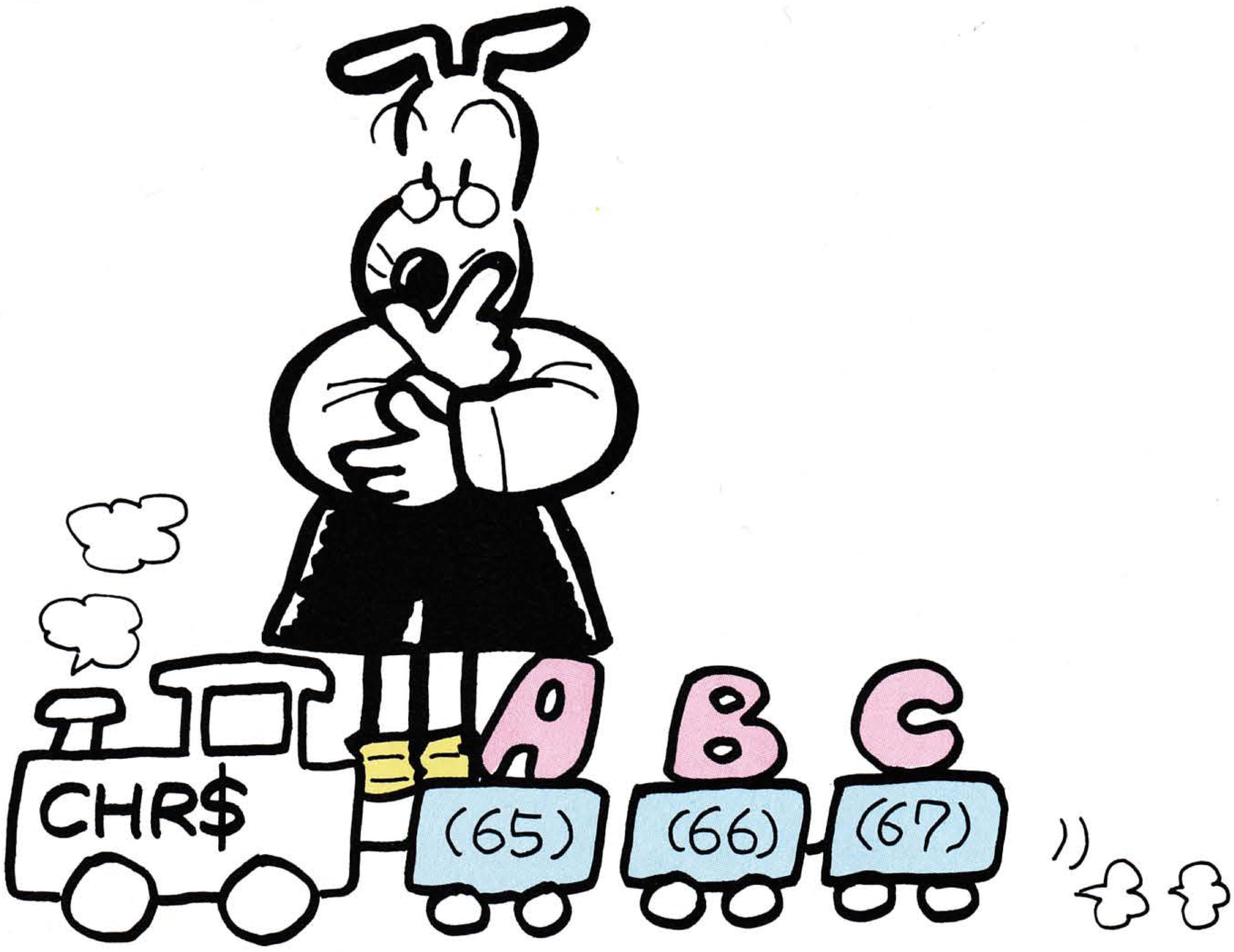
IF	A=B	THEN	100	ELSE	200
	条件		命令または行番号		命令または行番号

・条件が成立すればＴＨＥＮの後ろの命令を実行するか、またはＴＨＥＮの後ろの行番号の行にジャンプします。
・条件が成立しなければＥＬＳＥの後ろの命令を実行するか、またはＥＬＳＥの後ろの行番号の行にジャンプします。
・ＥＬＳＥとその後ろの命令または行番号は、いっしょに省略できます。省略した場合に条件が成立しないときは、次の行に進みます。

170行のＩＦ文は、正解と入力された答えの両方の行と列が等しい場合は、200行にジャンプします。180行のＩＦ文は、行か列のどちらか片方だけが等しい場合とどちらも等しくない場合で、表示される文字列を変えています。

文字も数値で表せる?

ＢＡＳＩＣで使うすべての文字には、それぞれ異なる番号が割り当てられています。この番号を「キャラクタコード」（または「文字コード」）と呼びます。
下のプログラムは、キーボードから入力できるすべての文字を、文字コードを使って画面に表示させるプログラムです。



リストと解説

プログラム	解説
`1 'save"ｾﾞﾝﾓｼﾞ.bas"`	ファイルスペック
`100 SCREEN 1:WIDTH 29:CLS`	画面の初期化と消去
`110 FOR A=33 TO 253`	「！」～「ん」の文字コードをＡの値に
`120   PRINT CHR$(A);`	Ａの値を関数の引数にして文字を表示
`130 NEXT A`	Ａの値を１増やして110行に戻る
`140 PRINT`	行を変える（改行）
`150 FOR A=65 TO 95`	グラフィックキャラクタのコードをＡの値に
`160   PRINT CHR$(1)+CHR$(A);`	グラフィックキャラクタの表示
`170 NEXT`	Ａの値を１増やして150行に戻る
`180 END`	プログラムの実行を終了する

※140行のように、ＰＲＩＮＴ命令の後ろに何も付けないで実行させると、改行だけが行なわれます。

新しい命令の説明

キャラクタコード表

ＢＡＳＩＣでは、すべての文字を番号であらわせます。この番号を「キャラクタコード」と呼びます。

キャラクタコードは、170ページの**「キャラクタコード表」**に載っています。
この一覧表ではキャラクタコードを10進数ではなく、**16進数**であらわしています。16進数はＢＡＳＩＣでよく使いますから覚えておきましょう。

・10進数……１桁を０～９の10個の数字であらわす。何も記号を付けない。
・16進数……１桁を０～９と、10～15をあらわすＡ～Ｆの16個の数字であらわす。
※　10進数と区別するため、16進数の前には**「&Ｈ」**を付けます。

10進数の数字	0	1	2	3	…	9	10	11	12	13	14	15	16	17
16進数の数字	0	1	2	3	…	9	A	B	C	D	E	F	10	11

たとえば「Ｍ」のキャラクタコード「&Ｈ４Ｄ」は、10進数では16×４＋13で「77」になります。

CHR$（キャラクタ・ダラー）（関数）

数値であるキャラクタコードを対応する文字に変える関数です。

引数は当然キャラクタコードですが、10進数でも16進数でもかまいません。

例）

```
PRINT  CHR$ (77);CHR$ (&H4D)
MM
Ok
■
```

グラフィックキャラクタとヘッダ

キーボードの［GRAPH］を押しながら入力する「月」などの文字を、「グラフィックキャラクタ」と呼びます。

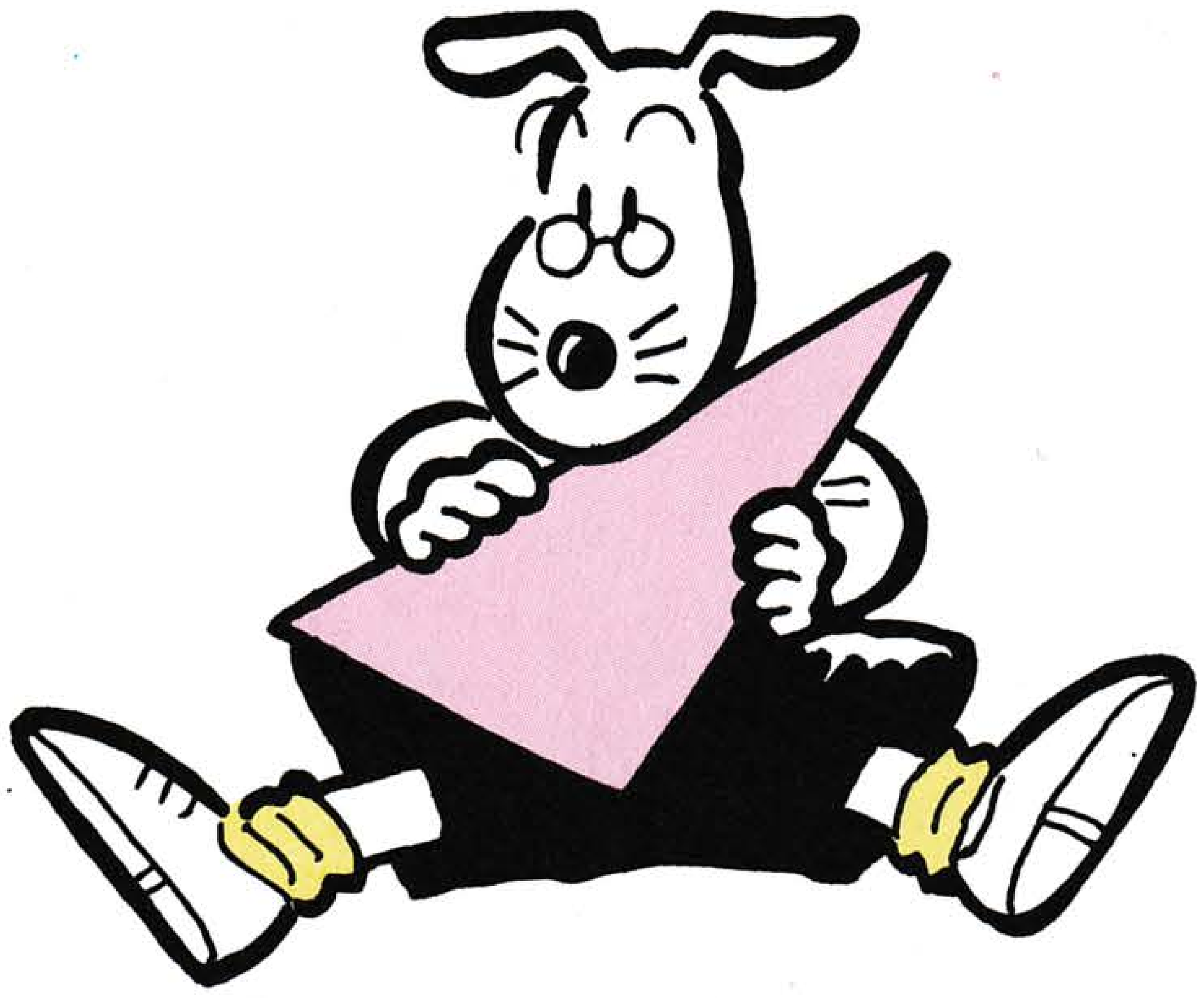
グラフィックキャラクタをキャラクタコードであらわすときは、キャラクタコードの前に**「グラフィックヘッダ」**と呼ばれる特別なコードを付けることが必要です。
グラフィックヘッダは、**「&Ｈ１」**に決められています。

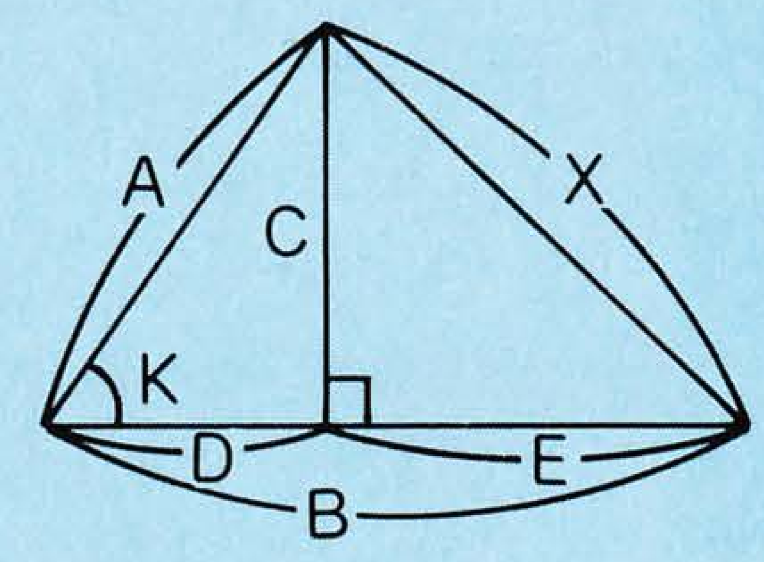
例）

```
PRINT  CHR$ (&H1) +CHR$ (&H4D)
百
Ok
■
```

謎の三角形

二辺の長さとそのあいだの角度が分かっている三角形の、残りの一辺の長さを求めるプログラムです。





リストと解説

```
1 'save"ｻﾝｶｸ.bas"
100 SCREEN 0:WIDTH 40:CLS
110 P=3.14
120 READ A,B,K
130 F=K*P/180
140 C=A*SIN(F)
150 D=A*COS(F)
160 E=B-D
170 X=SQR(C^2+E^2)
180 PRINT "A=";A;"B=";B;"K=";
K
190 PRINT "X=";X
200 END
210 DATA 5,7,60
```

- 100：画面の初期化（1行を40桁に）と消去
- 110：変数Pの値をπ（円周率）にする
- 120：ＤＡＴＡ文からデータを読み取り、各変数に
- 130：角度をラジアン単位に変換して変数Ｆに入れる
- 140：サイン関数で垂線Ｃの長さを求める
- 150：コサイン関数で辺Ｄの長さを求める
- 160：辺Ｅの長さを求める
- 170：目的の辺Ｘの長さを求める
- 180〜190：結果を表示する
- 200：プログラムの実行を終了する
- 210：辺Ａ，Ｂの長さと角度Ｋのデータ

実行すると、辺Ａの長さが５で辺Ｂの長さが７、その間の角度Ｋが60°の三角形の、残りの辺Ｘの長さを計算して三辺の長さを表示します。
210行の数値を書き換えて、いろいろな三角形の未知の辺の長さを求めることができます。

新しい命令の説明

READ DATA（リード、データ）

ＲＥＡＤは、プログラム中のＤＡＴＡ文から読み取ったデータを変数に入れる命令です。

必ずＤＡＴＡ文と組み合わせて使います。

```
120 READ A, B, K
         変数名 変数名 変数名
 ・
 ・
210 DATA 5, 7, 60
         データ データ データ
```

・データは、前から順に読み取られて変数の値になります。
・データの内容は、数値でも文字列でもかまいませんが、変数の型はデータと同じであることが必要です。
・ＲＥＡＤ文の変数の個数は、ＤＡＴＡ文のデータの個数以下でないとエラーになります。
・ＲＥＡＤ，ＤＡＴＡは、ダイレクトモードでは使えません。

SIN COS（サイン、コサイン）（関数）

ＳＩＮ，ＣＯＳは、ＢＡＳＩＣに用意されている組み込み関数のひとつで、数学の授業で習う三角関数のサイン・コサインと同じ働きをします。

例）

```
A=SIN (B)
変数名  引数

A=COS (B)
変数名  引数
```

ただし、次の点に注意してください。
・引数は、ラジアンを単位にします。（πラジアンで180°になります）

SQR（スクェアルート）（関数）

数学の授業で習う「√￣」（ルート）にあたる関数で、引数の平方根を求めるのに使います。

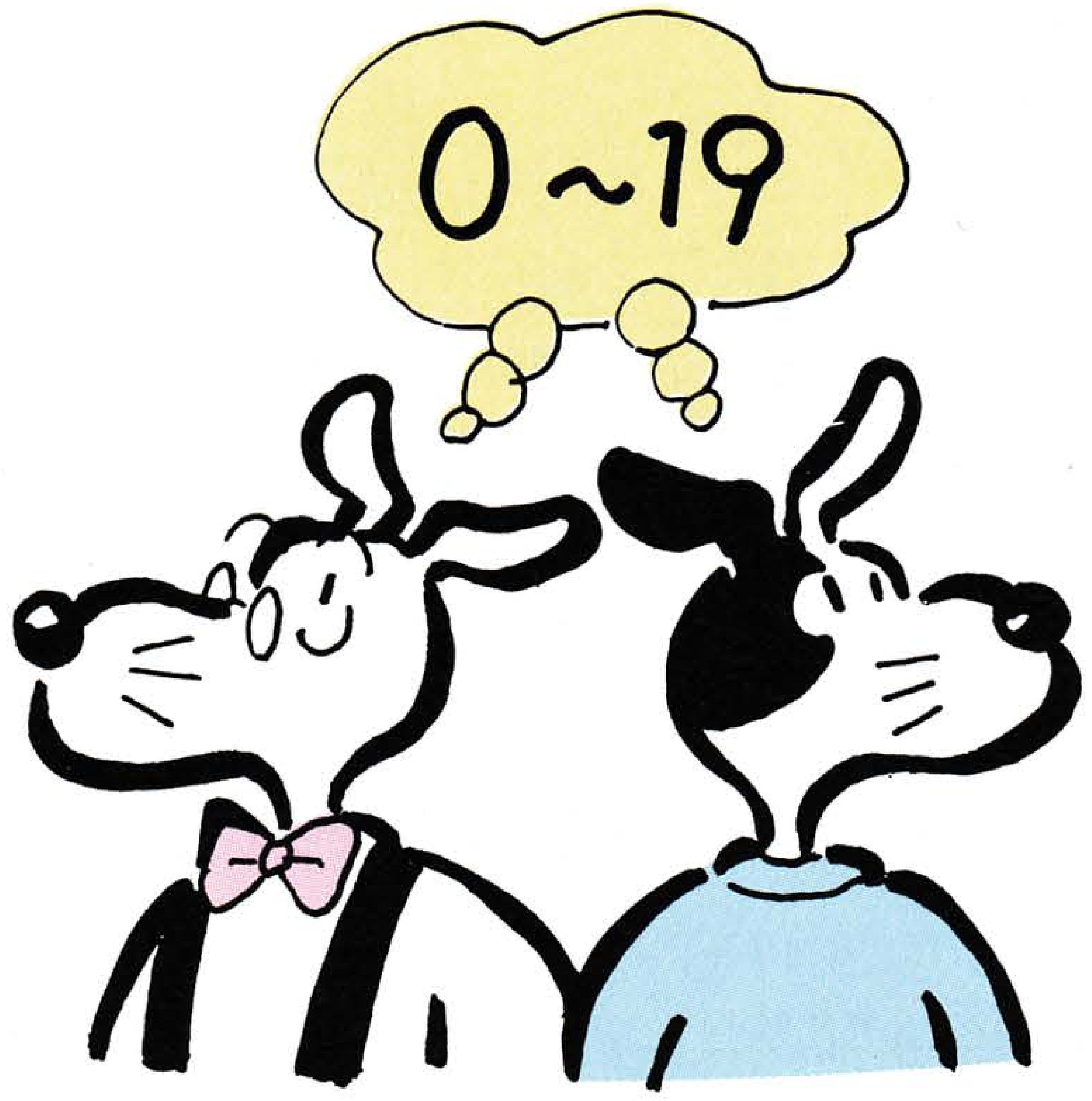
例）

```
A=SQR (B)
変数名  引数
```

記憶力ゲーム

二人で記憶力を競うゲームです。0～19までの数字を交互に１つずつ入力し、すでに入力されたのと同じ数字を入力してしまった人が負けです。自分と相手が入力した数字を覚えながら、できるだけデタラメな順番で数字を入力して相手を混乱させてください。

※メモを取ったりしたら反則ですよ！



リストと解説

```
1 'save"ｷｵｸﾘｮｸ.bas"
100 DIM M(19)
110 SCREEN 1:WIDTH 29:CLS
120 '
130 LOCATE 5,10:PRINT "
        "
140 LOCATE 5,10:INPUT "すうじ (0
-19)";N
150 IF N<0 OR N>19 THEN 120
160 IF M(N)=0 THEN M(N)=1:BEE
P:GOTO 120
170 '
180 FOR B=1 TO 10:BEEP:NEXT
190 LOCATE 7,12:PRINT"ざんねんで
した !"
200 END
```

- 100：M（0），M（1），…M（19）の20個の入れ物（配列変数）を用意する
- 110：画面の初期化と消去
- 130：空白を上書きして前回入力した数字を消す
- 140：数字の入力を求め、変数Nに入れる
- 150：数字が範囲を越えていたらやり直し
- 160：まだ入力されたことがない数字なら、配列変数に１を入れて120行へ　すでに入力された数字なら次の行へ
- 180：ビープ音を10回鳴らす
- 190：負けたことを表示する
- 200：プログラムの実行を終了する

新しい命令の説明

配列変数

プログラムの中で同じ種類のたくさんのデータを扱うとき、いちいち別の変数に入れるのは大変です。そんなときに便利なのが配列変数です。

普通の変数では、ひとつの変数名にはひとつの数値や文字列（データ）しか入れることができません。配列変数は、各データに番号をつけることによって、ひとつの変数名にたくさんのデータを入れることができます。
データを操作するときには、配列変数名の後ろに操作したいデータの番号を付けて指定します。この番号を**「配列変数の添え字」**と呼びます。

例）　Ｍ（６）　Ｂ＄（１０）　Ｃ３（Ｘ）

添え字を変数にしてプログラムの中で値を変えることにより、操作するデータの指定を変えることができます。

DIM（ディメンジョン）

配列変数を使用するときは、あらかじめ添え字の最大値を定義することが必要です。

そのための命令がＤＩＭです。

```
100 DIM M(19)
```

配列変数名　添え字の最大値

・ＤＩＭ文は、できるだけプログラムの先頭で実行させてください。
　プログラムの途中で実行させるとエラーになることがあります。
・１つのプログラム中で、同じ変数名でのＤＩＭ文の実行は、１回だけしかできません。

おまけのプログラム

ここまでガンバッテきた人にごほうびです。
いわゆる「ツルカメ算」の問題を解くプログラムです。

・使いかた

実行させると、まずツルとカメの頭の数の合計の入力を求め、次に足の数の合計の入力を求めます。

正しい数が入力されると、ツルとカメのそれぞれの数を表示します。
入力された数が正しくないときは、再び入力を求めます。

```
1 ' save"ツルカメ.bas"
100 SCREEN 1:COLOR15,4,7:CLS
110 LOCATE 5,7
120 INPUT "あたまのかす゛は";A
130 LOCATE 5,9
140 INPUT "あし のかす゛は";B
150 IF A<0 OR B<0 THEN 100
160 IF B MOD 2>0 THEN 100
170 IF A<B/4 THEN 100
180 K=B/2-A
190 T=A-K
200 LOCATE 5,12:PRINT"ツルは";T
210 LOCATE 5,14:PRINT"カメは";K
220 END
```

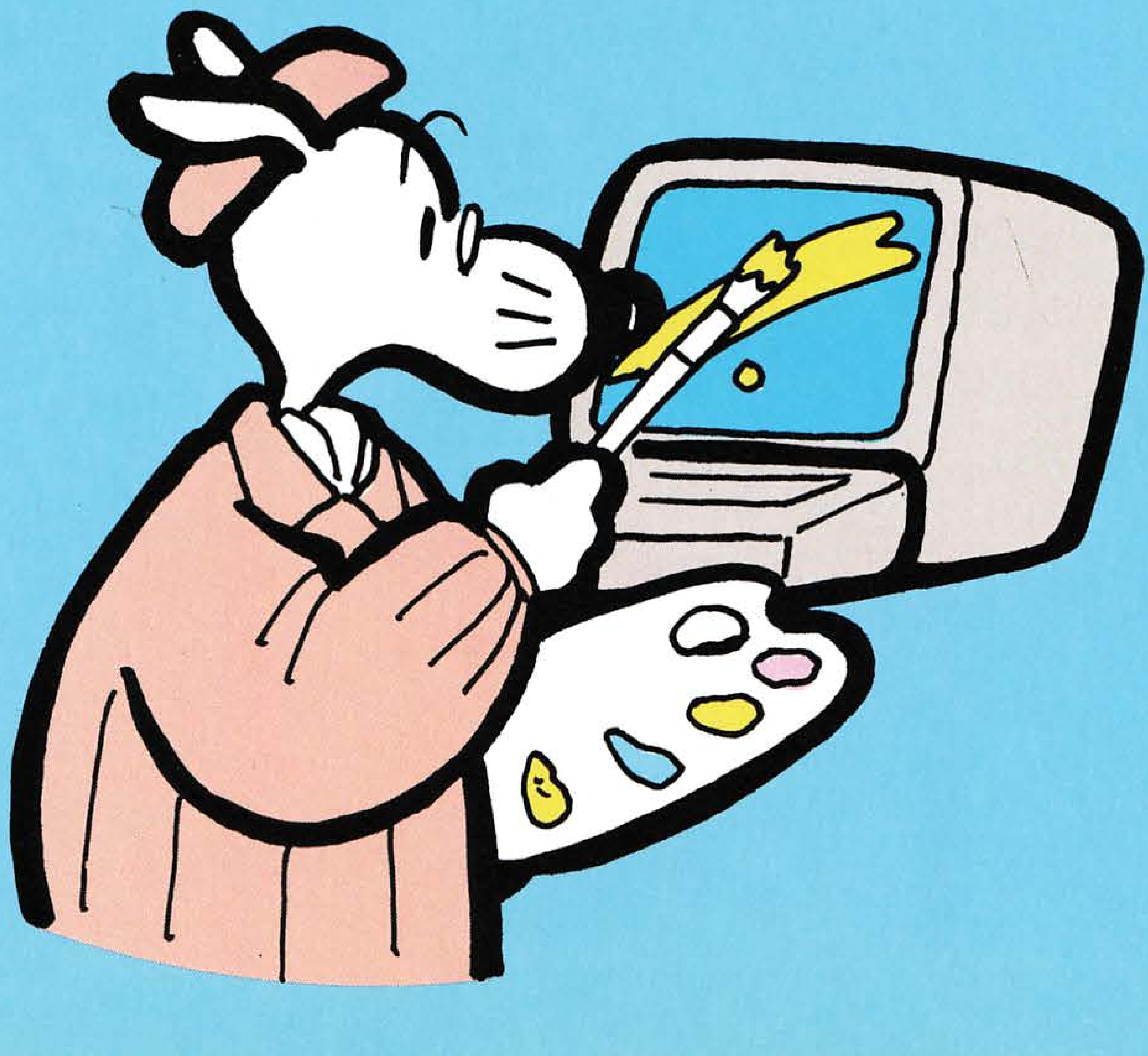
ご注意！ 学校の宿題は自分で解きましょう。

BASIC編

第3章

ゲームにはグラフィックが欠かせません。プログラムで画面に絵を描きましょう。

お待たせ!グラフィック



グラフィック画面へ

お待たせしました。ここからは、画面に絵を描くグラフィック関係の命令について説明します。
まず最初は、これまで使ってきた文字表示用のテキスト画面と絵を表示するためのグラフィック画面の違う点について説明します。

グラフィック画面の種類

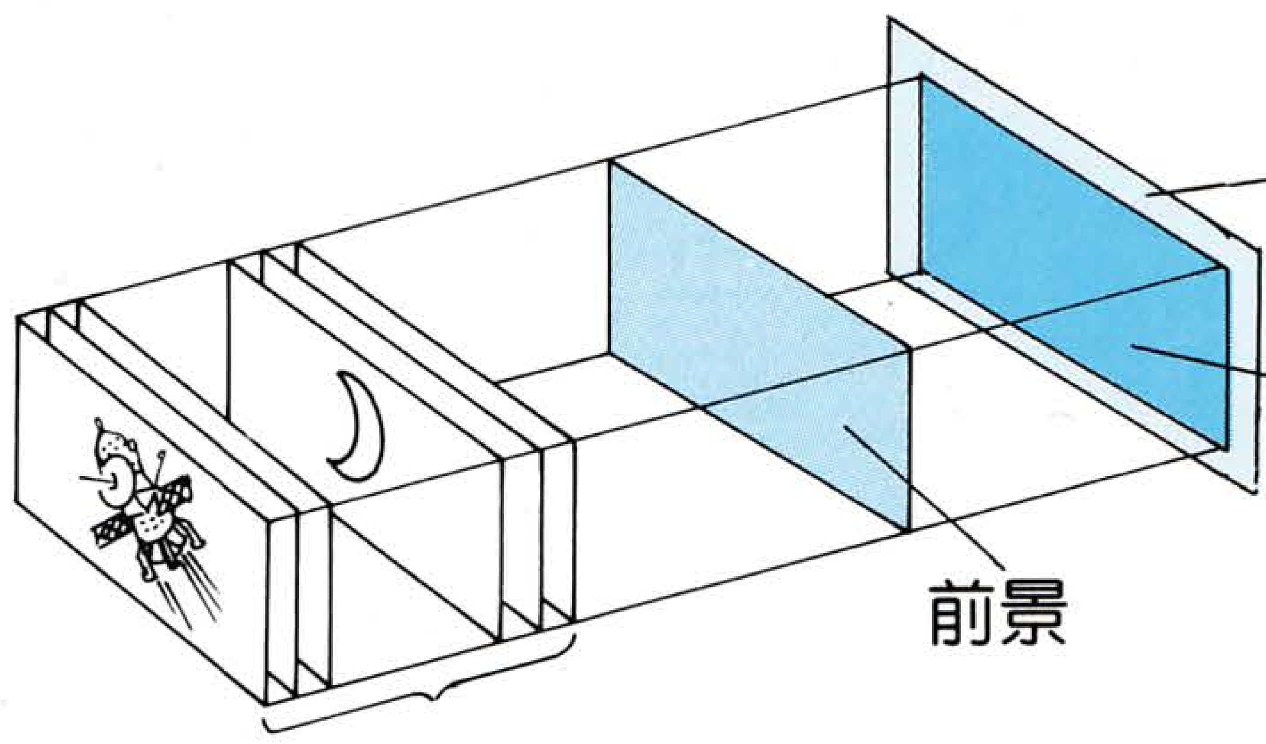
ＭＳＸ２+ パソコンの画面には、下の表のような種類があります。
（この表は、英数・かなモード時の各画面モードの仕様をしめしています）

モード	スクリーンNo.	文字表示	図形表示	表示色	色指定	スプライト
テキスト	0	初期値、1行39桁 最大80桁 ＊1	表示できない	512色中の16パレット中の2色	前景色 背景色 周辺色	表示できない
	1	初期値、1行29桁 最大32桁 ＊1	スプライトのみ可能（スプライト座標使用）＊2			8×8または16×16ドット 4個まで横方向の同じラインで同時表示可能
グラフィック	2	PRINT#文で英数かなのみ表示 ＊3	横256×縦192ドット	512色中の16パレット	水平8ドットを1単位とする領域ごとに2色	
	3		横256×縦192ドット（64×48ブロック）＊4		1ブロックごとに色指定可能	
	4		横256×縦192ドット		水平8ドットを1単位とする領域ごとに2色	8×8または16×16ドット 8個まで横方向の同じラインで同時表示可能
	5	PUT KANJI文で漢字表示可 ＊3	横256×縦212ドット		1ドットごとに色指定可能	
	6		横512×縦212ドット	512色中の4パレット		
	7			512色中の16パレット		
	8		横256×縦212ドット	256色固定		
自然画表示	10			自然画表示色、12,499色 ＊5		
	11	文字表示には向かない。		自然画表示色、19,268色 ＊5		
	12			自然画表示色、19,268色 ＊5		

＊1）英数・かなの表示数。
＊2）スプライト座標は、Xは−32〜255、Yは−32〜191（スクリーン5以上では、Yは−32〜211）。
＊3）漢字モードでは、全画面モードで漢字表示可能。 漢字モードについては、 98ページから説明。
＊4）1ブロックは4×4ドットの領域。
＊5）スクリーン11・12では、図形表示に256色分の色コードを指定できますが、色表示の方法が他の画面と異なるため、同じコードによる色指定をしても他の画面とは異なる色になったり、まったく表示されなかったりします。図形表示はスクリーン10までの画面モードでしてください。

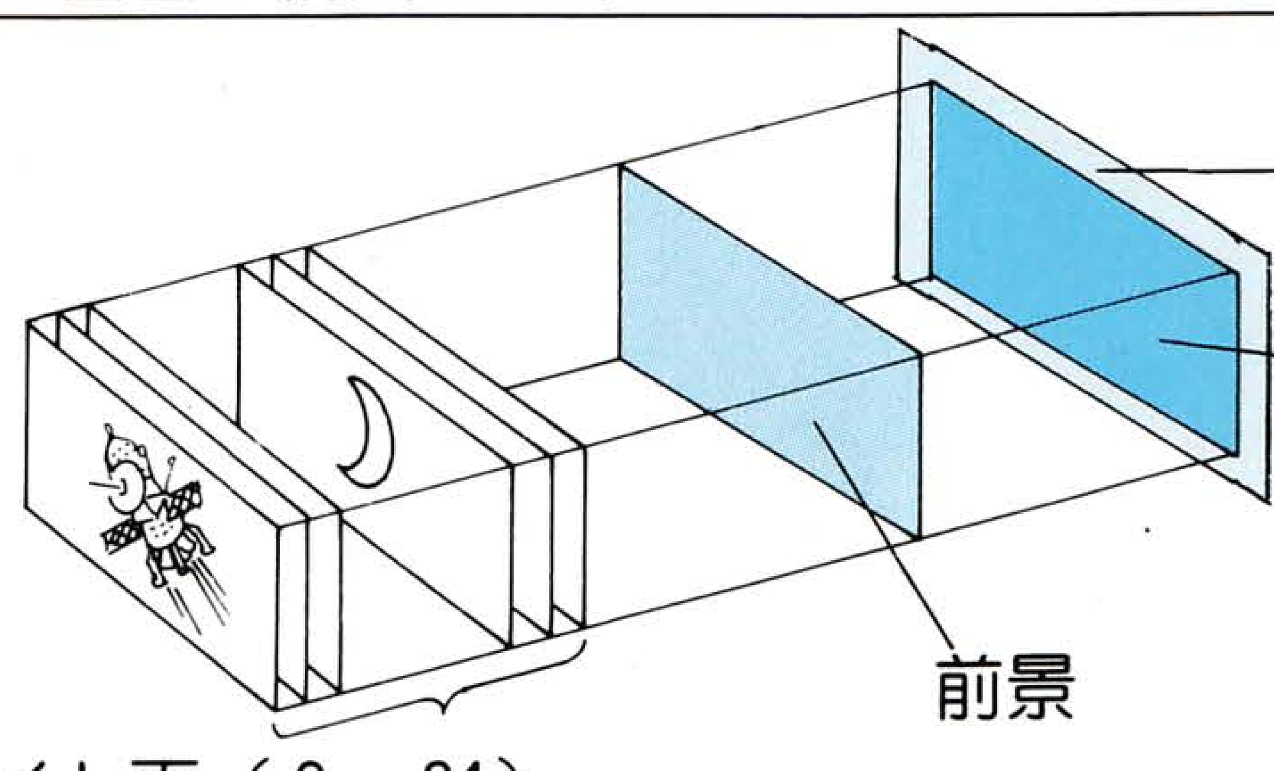
BASICの画面の構成

ＭＳＸ２+ パソコンの画面の種類を細かく分けると、スクリーン０～12の12種類になります。（スクリーン９はありません）これを、**「画面モード」**と呼びます。
スクリーン０～８の画面は、次のような構成になっています。



これらの名前はこれからの説明に出てきますから、覚えておいてください。
・ただしスクリーン０には、スプライト面はありません。

スクリーン10～12の画面の構成は、下のようになっています。



・**「自然画」**は、カラー写真などを特殊な装置を使って画像データに変換した絵のことです。スクリーン10～12の**「自然画表示面」**は、自然画を表示するための特別な方式による画像表示を行う面です。
・ＢＡＳＩＣでは自然画を作成することができません。
・スクリーン10で図形や文字を描くと、自然画表示面に表示されます。
・スクリーン11～12は自然画表示専用です。ＢＡＳＩＣで図形や文字を描くと、色が違ったり表示されなかったりします。

ＢＡＳＩＣがスタートしたときには、テキスト画面になっています。これをグラフィック画面に切り替えるには、次の命令を実行させます。

画面モードを切り替える(SCREEN)

ＳＣＲＥＥＮ（スクリーン）は、画面モードを切り替えたり、各種の設定を変更する命令です。

例）

ＳＣＲＥＥＮ　５
画面モード

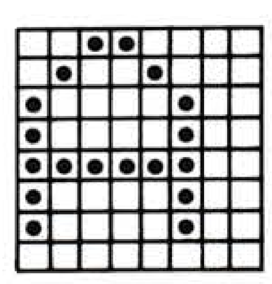
ＢＡＳＩＣがスタートしたときには、スクリーン１のテキスト画面が表示されます。
（ＳＥＴ　ＳＣＲＥＥＮ命令でスタート時の画面をスクリーン０に設定したときは除きます）

スクリーン０とスクリーン１の画面には、絵や図形を描くことができません。
これらの画面は**「テキスト画面」**と呼ばれ、文字の表示用です。
絵や図形を表示することができる画面を**「グラフィック画面」**と呼び、スクリーン２～12がそれにあたります。そのうちスクリーン10～12はＭＳＸ２＋パソコン特有の画面モードで、特に**「自然画表示モード」**と呼びます。

例）　ＳＣＲＥＥＮ　０　　（スクリーン０のテキスト画面に切り替える）
　　　ＳＣＲＥＥＮ　５　　（スクリーン５のグラフィック画面に切り替える）

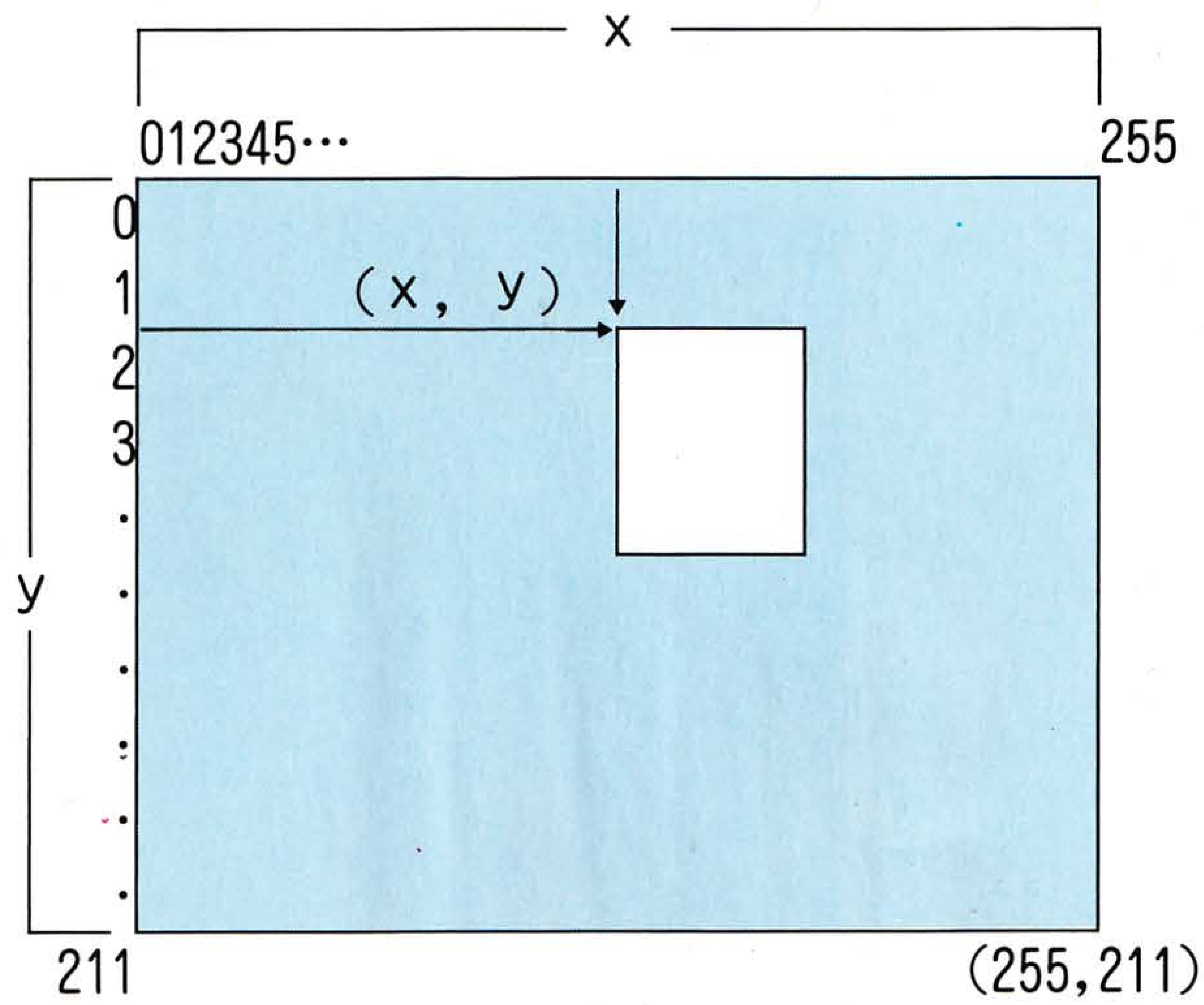
グラフィック画面での表示位置

テキスト画面、グラフィック画面の両方とも、画面に表示される文字や絵は、すべて「ドット」と呼ばれる小さな点を並べて表示されています。



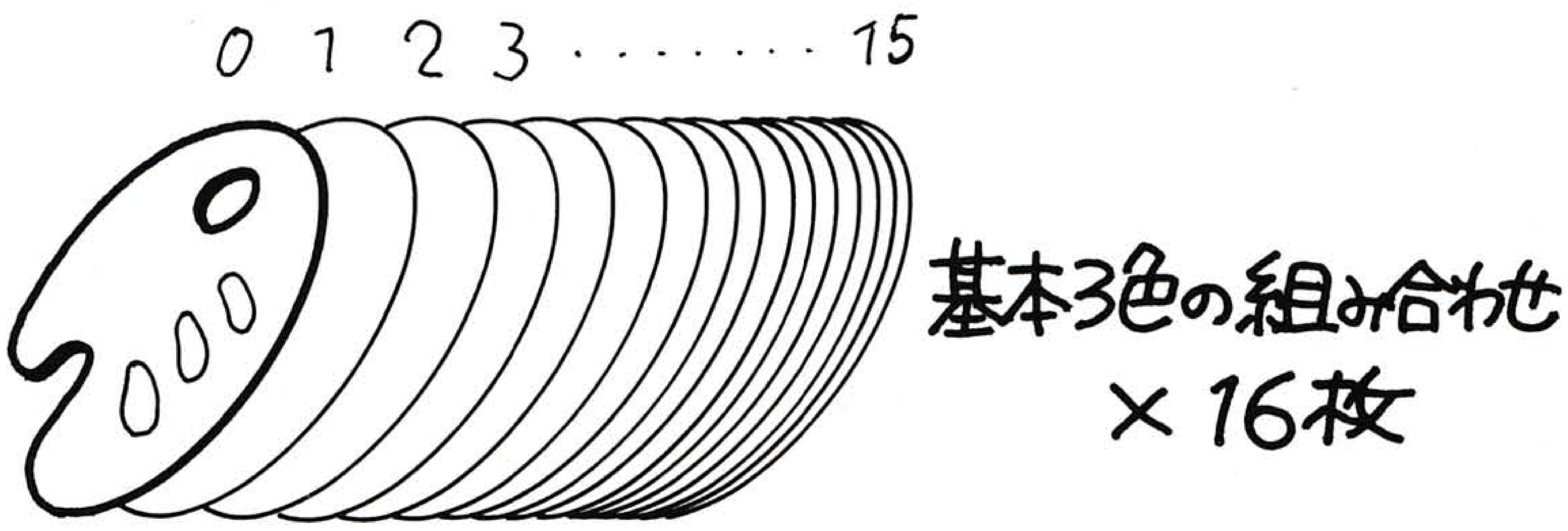
テキスト画面では文字を表示する位置を指定するのに、「ヨコ何桁目、タテ何行目」というように１文字単位で指定しましたが、グラフィック画面では点を表示する位置を指定するのに「ヨコ何ドット目、タテ何ドット目」と、１ドット単位で指定します。各画面モードがタテ方向とヨコ方向にそれぞれ何ドットあるかは、62ページの表の「図形表示」に載っています。

例）スクリーン５の位置座標



色の指定のしかた

スクリーン２〜７で使用できる色の種類は、 512色あります。ただし、そのうち同時に表示できるのは、最大16色までです。（スクリーン６では４色まで。）
ちょうど油絵や水彩画を描く人が絵の具をパレットの上で混ぜ合わせて使いたい色を作り出すように、０〜15までの番号に使いたい色を割り当て、実際に表示する時はこの番号で指定します。
この機能を**「パレット機能」**と呼び、番号を**「カラーコード」**と呼びます。つまり、スクリーン２〜７までの画面には16枚のパレットがあるわけです。

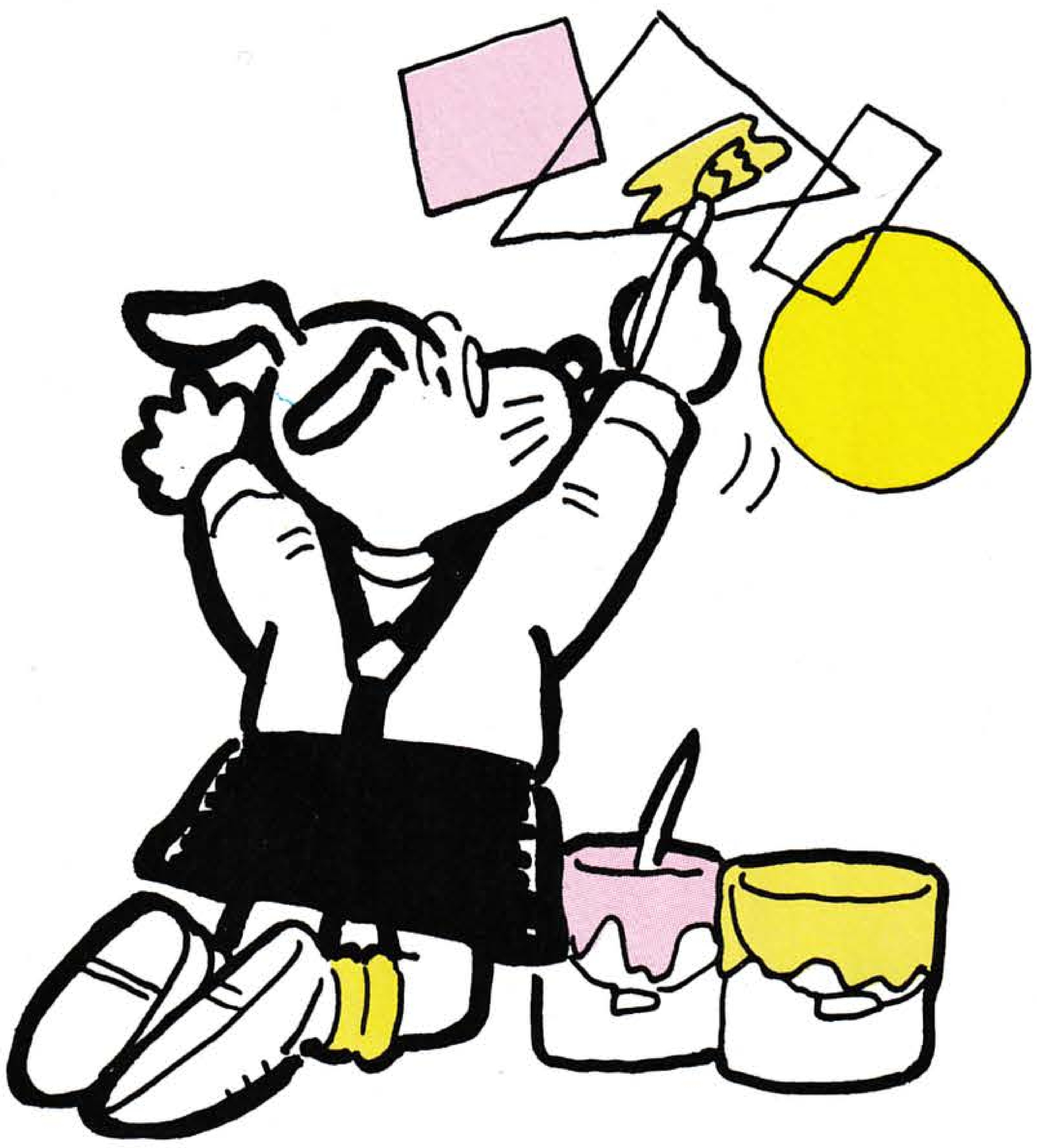


ＢＡＳＩＣがスタートしたときには各カラーコードに下の色が設定されますので、これらの色だけを使うときは、あらためて各パレットに色を設定する必要はありません。

		赤 緑 青の各成分		
カラーコード	０：透明	（０，０，０）	８：赤	（７，１，１）
	１：黒	（０，０，０）	９：淡い赤	（７，３，３）
	２：緑	（１，６，１）	10：黄	（６，６，１）
	３：淡い緑	（３，７，３）	11：淡い黄	（６，６，４）
	４：濃い青	（１，１，７）	12：濃い緑	（１，４，１）
	５：青	（２，３，７）	13：紫	（６，２，５）
	６：濃い赤	（５，１，１）	14：灰	（５，５，５）
	７：水色	（２，６，７）	15：白	（７，７，７）

直線、四角、円

それでは、実際にグラフィック画面に図形を描いてみましょう。



リストと解説

```
1 'save"ﾁｮｸｾﾝ.BAS"
100 SCREEN 5:COLOR:CLS
110 LINE(100,10)-(150,60),10
120 LINE( 30,10)-( 80,60),15,
B
130 LINE(170,10)-(220,60), 3,
BF
140 FOR C=1 TO 15
150   X=30:Y=70+C:D=4:D1=-D
160   PRESET (X,Y)
170   FOR M=0 TO 47
180     X=X+D:D1=-D1:Y=Y+D1
190     LINE -(X,Y),C
200   NEXT M
210 NEXT C
220 FOR R=10 TO 55 STEP 5
230   CIRCLE (128,150),R,2
240 NEXT R
250 PAINT (55,35),8,15
260 GOTO 260
```

行	解説
100	画面モードをスクリーン5に切り替える
110	カラーコード10（黄）で直線をひく
120	カラーコード15（白）で四角を描く
130	カラーコード3（淡い緑）で四角を描き、中を塗りつぶす
140	コードを指定する変数の値を15回変える
150	ジグザグを描き始める位置と変化値
160	各線を最初に描き始める位置を指定する
170	直線を描くのを48回繰り返す
180	直線を描き終える位置の値を変える
190	前回の位置から今回の位置まで直線を描く
220	円の半径を5ずつ増しながら繰り返す
230	カラーコード2（緑）で円を描く
250	カラーコード15（白）で囲まれた部分を8の色（赤）で塗りつぶす
260	テキスト画面に戻らないようにする

新しい命令の説明

LINE（ライン）

ＬＩＮＥは、グラフィック画面に直線や四角を描く命令です。

直線を描くとき

```
110 LINE(100,10)-(150,60),10
```
始点の位置座標　終点の位置座標　カラーコード

四角を描くとき……・・B

```
120 LINE( 30,10)-( 80,60),15,B
```
対角をなす点の位置座標　四角形の指定

四角形を描き、同じ色で中を塗りつぶすとき……・・BF

```
130 LINE(170,10)-(220,60), 3,BF
```
塗りつぶしの指定

PRESET（ポイント・リセット）

ＰＲＥＳＥＴは、描いた点を消すための命令です。

```
160 PRESET (X,Y)
```
ヨコ座標　タテ座標

このプログラムでは、次のＬＩＮＥ命令で線を描き始める位置（最終参照点）を設定するために使われています。（上の例の「Ｘ」,「Ｙ」は変数名です。）

CIRCLE（サークル）

ＣＩＲＣＬＥは、グラフィック画面に円を描く命令です。

```
230 CIRCLE (128,150),R,2
```
中心の位置座標　半径　カラーコード

・半径はドット単位で指定します。（上の例の「Ｒ」は変数名です。）

PAINT（ペイント）

ＰＡＩＮＴは、指定した位置を含む、境界色で囲まれた部分を領域色で塗りつぶす命令です。

```
250 PAINT (55,35),8,15
```
領域内の位置の座標　領域色　境界色

・指定した位置が境界色で完全に囲まれていないときは、画面全体が領域色で塗りつぶされます。

いろいろな色

ＢＡＳＩＣがスタートしたときに各カラーコードに割り当てられる色を画面で確かめるプログラムです。「ー」（マイナス）の数値を入力すると、パレット機能の効果を確かめることができます。



実行させるとカラーコードの入力を求めます。０～15までの数値を入力すると、画面の周辺領域の色が変わります。（ただし、コード０は透明、コード１は黒）
マイナスの数値を入力すると、コード13の赤、緑、青各成分の入力を順に求めます。
０～７までの数値を入力してください。
表示される文字の色（コード13）が変化します。

プログラムを止めるときは、［CTRL］を押しながら［STOP］を押してください。

リストと解説

```
1 'save"イロイロなイロ.bas"
100 SCREEN 1:WIDTH 29:C1=7
110 COLOR:COLOR 15,4,C1:CLS
120 LOCATE 5,11:INPUT "COLOR ";C1
130 IF 15< C1 THEN 120
140 IF  0<=C1 THEN 110
150 CLS:COLOR 13,4,7
160 PRINT:PRINT "color 13 の R = ";:INPUT R
170 PRINT "color 13 の G = ";:INPUT G
180 PRINT "color 13 の B = ";:INPUT B
190 IF R<0 OR 7<R THEN 160
200 IF G<0 OR 7<G THEN 160
210 IF B<0 OR 7<B THEN 160
220 COLOR= (13,R,G,B)
230 GOTO 160
```

- 100：画面を初期化し、変数Ｃ１に７を入れる
- 110：周辺色をＣ１の値にして画面を消去する
- 120：カラーコードの入力を求め、入力された値をＣ１に入れる
- 130：Ｃ１の値が15より大きければ、再入力を求める
- 140：Ｃ１の値が０以上なら、110行に戻る
- 150：Ｃ１の値が０以下なら、前景色をコード13に
- 160：カラーコードの13の赤成分の入力を求め、変数Ｒに入れる
- 170：カラーコードの13の緑成分の入力を求め、変数Ｇに入れる
- 180：カラーコードの13の青成分の入力を求め、変数Ｂに入れる
- 190～210：Ｒ，Ｇ，Ｂ各成分の値が０～７の範囲外なら、色を変更しないで再入力を求める
- 220：コード13の色を入力された成分比に変更する
- 230：160行に戻る

新しい命令の説明

COLOR
（カラー）

次のように入力すると、画面の前景色、背景色、周辺色を変更します。

```
110  COLOR 15, 4, C1
           前景色 背景色 周辺色
```

・テキスト画面、グラフィック画面のどちらでも使えます。
・各色はカラーコードで指定します。使えるカラーコードの範囲は、画面モードによって変わります。
・次の場合には、文字などの表示が見えなくなります。
　→前景色と背景色に同じコードを指定したとき
　→前景色にカラーコード0（透明）を指定したとき
・ＢＡＳＩＣがスタートしたときには、「COLOR 15,4,7」に設定されています。
　F6 （SHIFT を押しながら F1 を押す）を入力するとこの設定になります。
・後述の漢字テキスト画面でＣＯＬＯＲ文を実行するとエラーになります。漢字テキスト画面では、ＣＡＬＬ　ＰＡＬＥＴＴＥ文を使用してください。（⇨174ページ）

次のように入力すると、指定されたカラーコードの赤、緑、青各成分を指定された比率にして、実際に表示される色を変えます。

```
220  COLOR=(13, R, G, B)
     カラーコード，赤の比率，緑の比率，青の比率
```

・この指定は、スクリーン0～7および10で有効です。スクリーン8にはパレット機能はありません。またスクリーン11～12では、指定した比率の色になりません。
・カラーコードは1～15までです。（スクリーン6では1～3）
・カラーコード0（透明）は、ＣＯＬＯＲ文では変更できません。
・赤，緑，青の各比率は、0～7までの数値で指定します。

スクリーン8は宝石箱

スクリーン８にはパレット機能はありませんが、そのかわり、 256色まで同時に表示する機能があります。この機能を使って、たくさんの色で画面に絵を描いてみましょう。（このプログラムでは、あわせて、英数・かなモードのグラフィック画面に漢字を表示する機能も使っています）



プログラムを止めるときは、 CTRL を押しながら STOP を押してください。

※漢字モードでの漢字の表示のしかたは、98ページから説明します。

リストと解説

```
1 'save"ﾎｳｾｷﾊﾞｺ.bas"
100 SCREEN 8:COLOR 255,0
110 LINE (50,100)-(150,160),2
55,BF
120 LINE (50,100)-(100, 50)
130 LINE          -(200, 50)
140 LINE          -(150,100),1
82
150 LINE          -(150,160),1
82
160 LINE          -(200,110),1
82
170 LINE          -(200, 50),1
82
180 PAINT (175,100),182
```

- 100：スクリーン８に：前景色を白に、背景色を透明に
- 110：カラーコード 255（白）で宝石箱の前面を描く
- 120〜130：宝石箱の上面のへりを描く
- 140〜170：宝石箱の側面をカラーコード182で描く
- 180：同じ色で側面を塗りつぶす

```
190 PUT KANJI (76,122),&H4A75
,28
200 PUT KANJI          ,&H4050
,28
210 PUT KANJI          ,&H4822
,28
220 A=RND(-TIME)
230 '
240 Y=INT(51+RND(1)*49)
250 X=151-Y+RND(1)*99
260 C=256*RND(1)
270 PSET (X,Y),C
280 GOTO 240
```

- 190〜210：漢字コード&H4A75,&H4050,&H4822の漢字（「宝」「石」「箱」）を赤で表示する
- 220：乱数系列を初期化する
- 240：Yに51〜99の範囲の乱数を入れる
- 250：Xに-46〜100の範囲の乱数を入れる
- 260：Cに0〜255の範囲の乱数を入れる
- 270：X，Yの位置にCの色で点を描く
- 280：240行に戻る

新しい命令の説明

スクリーン8

スクリーン8のカラーコードは、表示される色を次のような形式で直接にあらわしています。

緑色の成分……G（0〜7の8段階）
赤色の成分……R（0〜7の8段階）
青色の成分……B（0〜3の4段階）

計算方法

G×32＋R×4＋B＝カラーコード

したがって、スクリーン8で指定できるカラーコードは、0〜255の256種類になります。

・カラーコード0は、他の画面モードと同じ透明です。
・カラーコード255は、白です。

PUT
KANJI
(プット
・カンジ)

ＰＵＴ　ＫＡＮＪＩは、英数・かなモードで漢字をグラフィック画面に表示する命令です。

```
190 PUT KANJI (76, 122), &H4A75, 28
                ヨコ座標 タテ座標  漢字コード カラーコード
```

・グラフィック画面のスクリーン5〜10でだけ使用します。

・ヨコ座標、タテ座標は、文字の左上の位置をあらわします。文字はタテ16ドット，ヨコ16ドットの範囲の中に表示されます。省略したときは、最後に実行した位置座標を使うグラフィック関係の命令で使用した位置（最終参照点）に表示されます。
ＰＵＴ　ＫＡＮＪＩを実行すると、最終参照点は16ドット右に移動します。

・**漢字コード**は取扱説明書の「ＪＩＳ漢字コード表」に載っています。（ただし、非漢字については、本書の205ページを参照のこと）漢字コード表は「キャラクタコード表」と同様に、ＪＩＳコードが16進数で表示されています。16進数は、前に「&Ｈ」を付けて入力します。

・カラーコードは、使用する画面モードによって使えるコードの範囲が変わります。
カラーコードを省略すると、前景色で表示されます。

・表示される文字は、文字の形の描線とそれを囲む四角い部分で構成されています。

PSET
(ポイント
・セット)

ＰＳＥＴは、グラフィック画面に点を表示する命令です。

```
270 PSET (X, Y), C
         ヨコ座標 タテ座標 カラーコード
```

・カラーコードは省略できます。省略すると、カラーコードは前景色になります。

・ＰＳＥＴを実行すると、最終参照点がその位置に移動します。

《テクニック》

７０ページのサンプルプログラムでは、漢字が表示された部分の周囲が黒くなってしまいます。これはグラフィック画面に表示される文字が、文字の部分とその周囲の四角い部分でできているからです。文字の周囲の部分は、自動的に背景色（この場合は透明）になります。
文字の周囲を背景色にしたくないときは、次のようにプログラムを変更します。

```
190 PUT KANJI (76, 122), &H4A75, 28, TPSET
200 PUT KANJI           , &H4050, 28, TPSET
210 PUT KANJI           , &H4822, 28, TPSET
```

プログラムを実行すると、箱の絵の上に漢字の文字の部分だけが表示されます。

ＴＰＳＥＴは**「論理演算子」**と呼ばれるもののひとつで、ＰＵＴ　ＫＡＮＪＩ命令といっしょに使用すると、透明色（カラーコード０）の部分を表示させないようにします。

したがってＴＰＳＥＴを使用するときは、あらかじめ背景色をカラーコード０に設定してください。

レモンを描こう

画面に半分に切ったレモンを描くプログラムです。
だ円や扇形を描くＣＩＲＣＬＥ命令の便利な機能を使っています。

プログラムを止めるときは、 CTRL を押しながら STOP を押してください。



リストと解説

```
1 'save"ﾚﾓﾝ.BAS"
100 P=3.14:X0=160:Y0=100
110 SCREEN5:COLOR 15,4,7:CLS
120 LINE (0,120)-(255,211),2,
BF
130 CIRCLE (X0,Y0),120,10,-.5
*P,-1.5*P, .75
140 PAINT  (X0-1,Y0),10
150 CIRCLE (X0,Y0),90,15,,,2.
5
160 PAINT  (X0,Y0),15
170 CIRCLE (X0,Y0),80,10,,,2.
5
180 PAINT  (X0,Y0),10
190 FOR J=-2*P TO 0 STEP .8
200   CIRCLE(X0,Y0),80,15,J,J
+.4,2.5
210 NEXT J
220 GOTO 220
```

行	解説
100	変数Ｐに円周率を、Ｘ0，Ｙ0に円の中心座標を
110	画面をスクリーン５にし、初期設定
120	緑色でテーブルの表面を描く
130	黄のだ円でレモンの胴体を描く
140	胴体を黄で塗る
150	白のだ円でレモンの断面を描く
160	断面を白く塗る
170	黄のだ円でレモンの中身を描く
180	中身を黄で塗る
200	白でレモンの房をあらわす扇形を描く

新しい命令の説明

CIRCLE
命令による
だ円や扇形
の描きかた

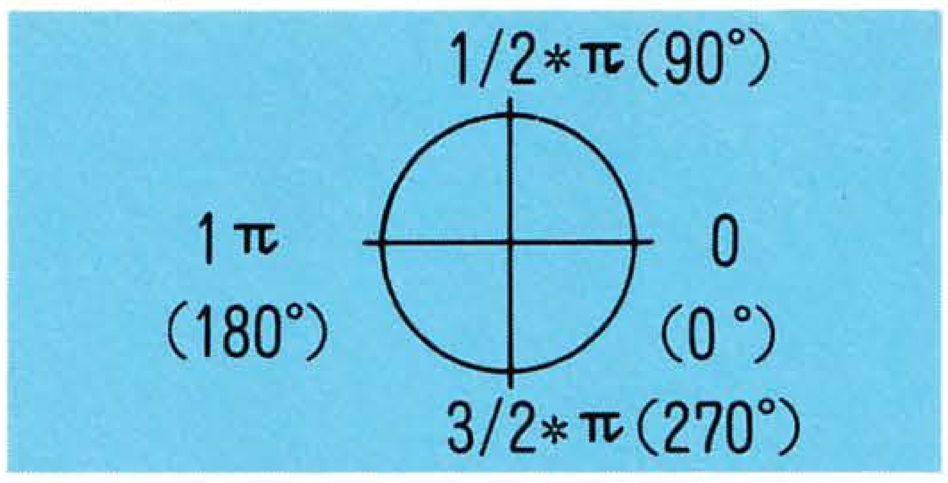
これまで説明を省略しましたが、ＣＩＲＣＬＥ命令には、だ円や扇形を描く機能もあります。

```
130 CIRCLE (X0,Y0),120,10,-.5*P,-1.5*P,.75
```

(X0,Y0)	120	10	-.5*P	-1.5*P	.75
中心の位置座標	半径	表示色	開始角度	終了角度	比率

新しく出てきた項目について説明します。

・**開始角度，終了角度**は、円弧を描くときに入力します。角度の単位はラジアンです。180°が1π、360°が2πになります。（π=3.14）



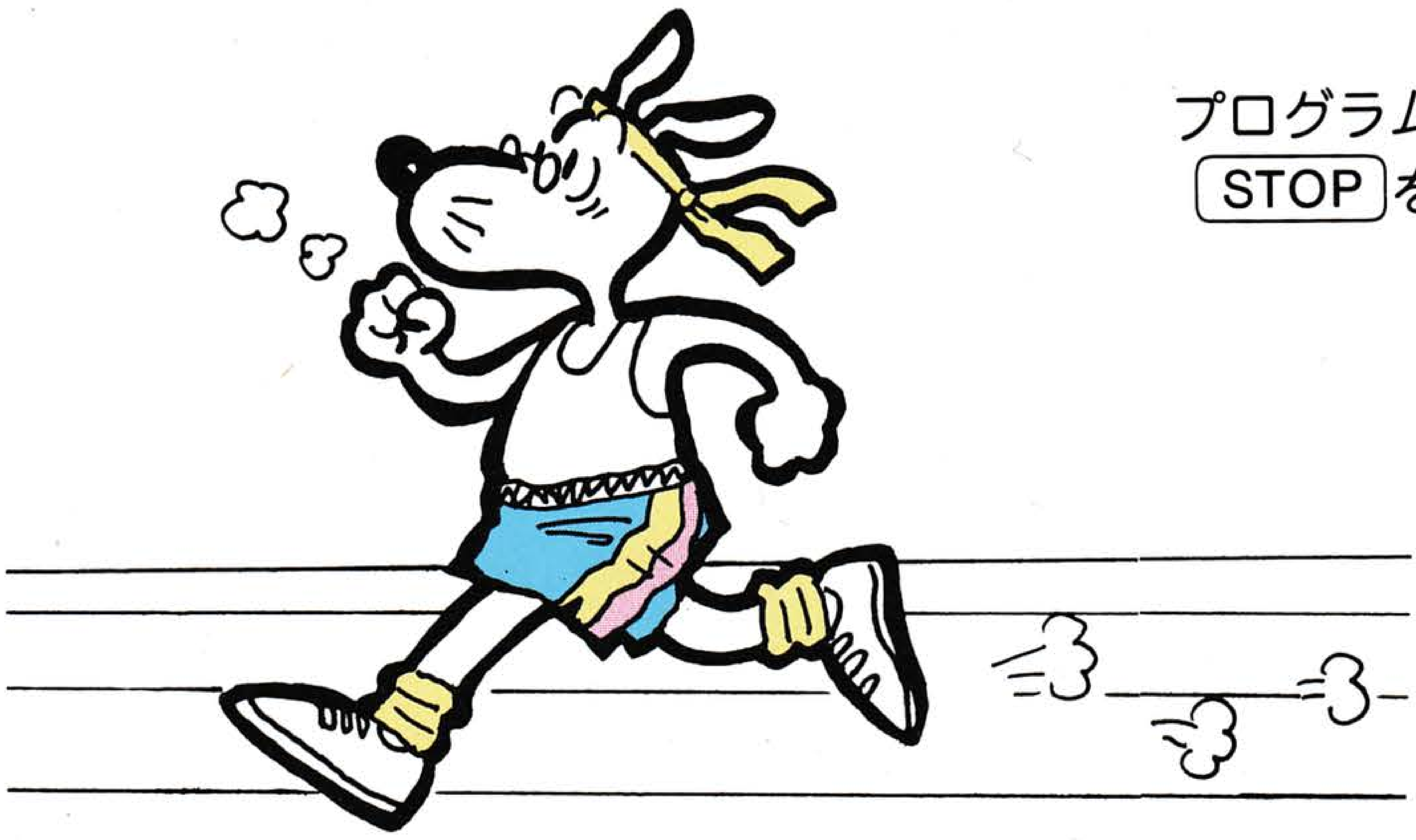
※常に円の右端が0になります。
※0から反時計回りに開始角度、終了角度の順に指定します。

開始角度，終了角度の数値の範囲は、−2π〜+2πまでです。
開始角度を省略したときは0、終了角度を省略したときは2πになります。
（途中の項目を省略するときには、「，」は省略しないで入力してください）
開始角度，終了角度の値をマイナスの数値にすると、プラスの数値のときと同じ円弧を描いたあと、中心から円弧の両端に直線が描かれ、扇形になります。

・**比率**は、だ円を描くときに入力します。（タテ方向の半径／ヨコ方向の半径）の数値で入力します。数値が1以上ならタテに長いだ円になり、1以下ならヨコに平たいだ円になります。省略したときは、1になります。

スプライトって何だろう

スプライトは、画面に表示されている絵や図形の上に重ねて表示し、速く移動させることができます。また、移動させるときには同じスプライト面の新しい位置にスプライトを表示するだけで前のスプライトが自動的に消える便利な機能をもっています。



プログラムを止めるときは、［CTRL］を押しながら［STOP］を押してください。

リストと解説

```
1 'save"ｽﾌﾟﾗｲﾄ.bas"
100 SCREEN 2,1
110 '--- ｽﾌﾟﾗｲﾄ ﾊﾟﾀｰﾝ ﾉ ﾃｲｷﾞ
120 FOR J=0 TO 4
130   B$=""
140   FOR K=1 TO 8
150     READ A$:B$=B$+CHR$(VA
L("&H"+A$))
160   NEXT K
170   SPRITE$(J)=B$
180 NEXT J
190 '* ﾊｲｹｲ ｦ ｴｶﾞｸ *
200 COLOR 15,4,7:CLS
210 FOR N=1 TO 10
220   X=N*N+95
230   LINE (0,X)-(256,X),12
240 NEXT N
```

- 100：画面をスクリーン2に、スプライトの種類を1（8×8ドット拡大）に設定
- 120：スプライトパターン5個を定義する
- 130：文字型変数B$の値を初期化する
- 140：8×8ドットパターンのデータを8回読む
- 150：DATA文から読み取ったデータをA$に入れ16進数にして文字列に変換、B$の値に連結する
- 170：B$の値をJ番のスプライトのパターンとして定義する
- 200：画面の初期設定と消去
- 210～240：左右に直線を10本描く

`250 '**** ｼﾞｯｺｳ ﾙｰﾁﾝ ****`	
`260 X=120:Y=90:G=0:T=0:PLAY "S9M400T250"`	表示位置や状態をあらわす変数を初期設定し、音色を設定する
`270 '`	
`280 PUT SPRITE 0,(X,Y),10,T*2+G:PUT SPRITE 1,(X,Y),13,4`	人物の体のスプライトと衣服のスプライトを重ねて表示する
`290 NX=50+G*10:PLAY "L16N=NX;"`	音を鳴らす
`300 FOR M=0 TO 19:NEXT`	移動スピードを調節するためのタイマー
`310 '--- ｶｰｿﾙｷｰ`	
`320 S=STICK(0)`	どのカーソルキーが押されているか調べる
`330 IF S=3 THEN T=0:GOTO 360`	右向きのキーならTの値を0にして360行へ
`340 IF S=7 THEN T=1:GOTO 360`	左向きのキーならTの値を1にして360行へ
`350 GOTO 310`	どちらも押されていなければ、310行へ戻る
`360 '`	
`370 G=1-G`	姿勢を表すGの値を反対にする
`380 ON T+1 GOTO 390,420`	Tに1を加えた値により390または420行へ
`390 '--- T=0`	
`400 X=X+4:IF 240=<X THEN X=240`	ヨコ座標に4を加え、240以上なら240にする
`410 GOTO 270`	270行へ戻る
`420 '--- T=1`	
`430 X=X-4:IF  X=<0 THEN X=0`	ヨコ座標から4を引き、0以下なら0にする
`440 GOTO 270`	270行へ戻る
`450 '--- ｽﾌﾟﾗｲﾄ ﾊﾟﾀｰﾝ ﾃﾞｰﾀ`	
`460 DATA 18,19,66,80,0,A4,42,2`	人物が右向きで手足を開いたときのパターンのデータ
`470 DATA 18,18,20,14,0,18,28,28`	人物が右向きで手足を閉じたときのパターンのデータ
`480 DATA 18,98,66,1,0,25,42,40`	人物が左向きで手足を開いたときのパターンのデータ
`490 DATA 18,18,4,28,0,18,14,14`	人物が左向きで手足を閉じたときのパターンのデータ
`500 DATA 0,0,18,18,18,0,0,0`	人物の衣服のパターンのデータ

このプログラムを実行させると、画面に人物のスプライトが表示されます。左右のカーソルキーを使って、人物を走らせてみてください。

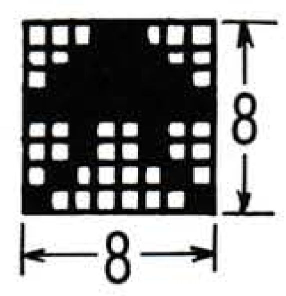
新しい命令の説明

スプライトの種類

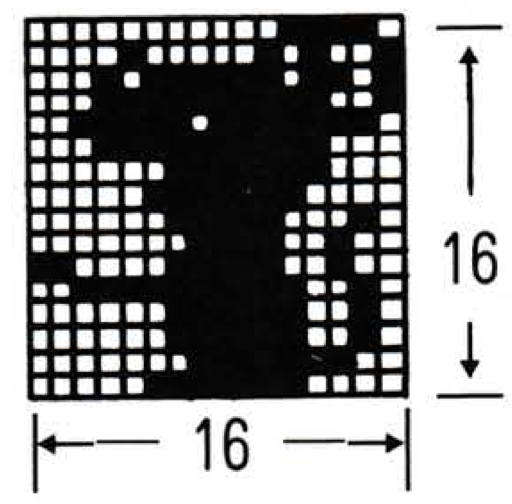
ＭＳＸ２+パソコンで使えるスプライトには、次の４種類があります。

スプライトの種類	表示される大きさ	使用できる画面モード	番号の範囲
0	8×8ドット標準	スクリーン1～12	0～255
1	8×8ドット拡大	スクリーン1～12	0～255
2	16×16ドット標準	スクリーン1～12	0～ 63
3	16×16ドット拡大	スクリーン1～12	0～ 63

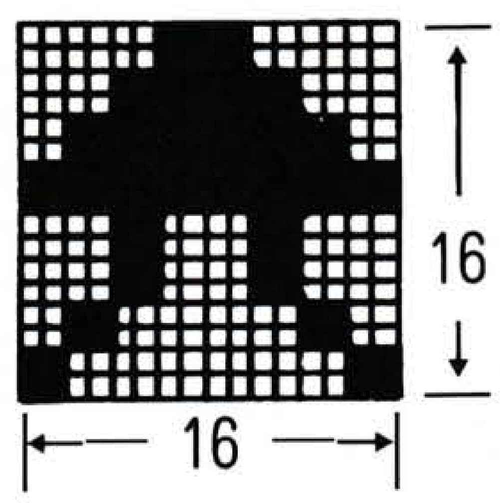
8×8ドット標準
（例：ＵＦＯ）



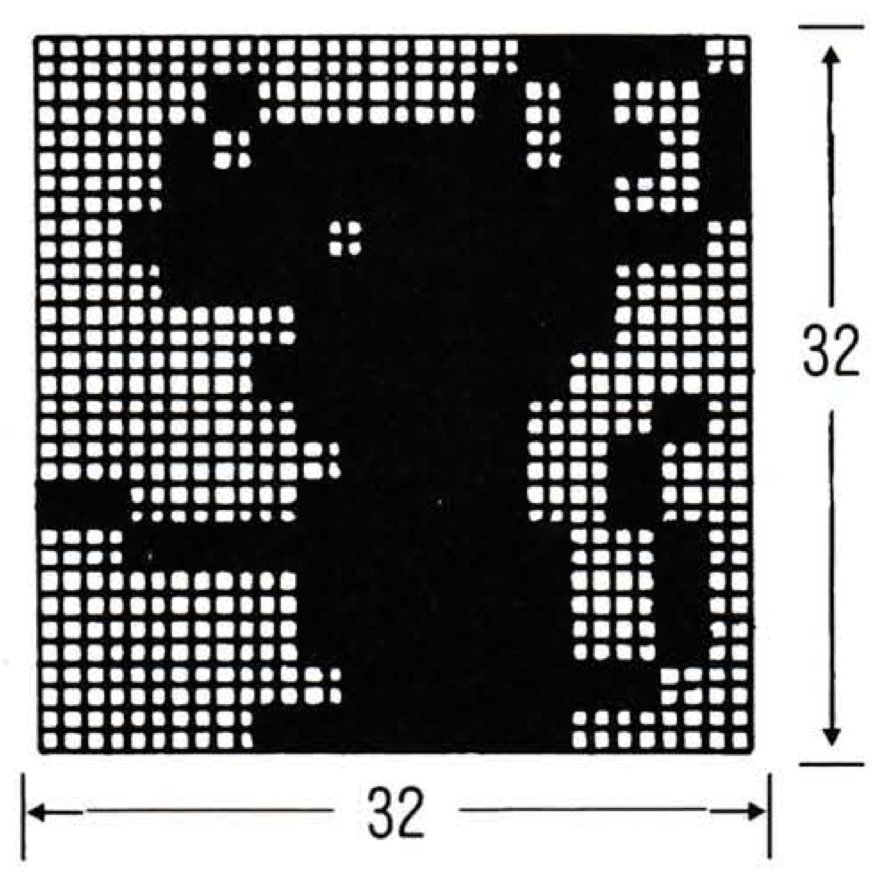
16×16ドット標準
（例：ねずみ）



8×8ドット拡大



16×16ドット
拡大



拡大は標準の大きさをタテ、ヨコとも2倍の大きさにしたものです。

SCREEN文によるスプライトの種類の指定

スプライトの種類は、ＳＣＲＥＥＮ文で指定します。

```
100 SCREEN 2, 1
```

画面モード　スプライトの種類

一度スプライトの種類を指定すると、すべてのスプライトが指定した大きさになります。
スプライトの種類を省略したときは、8×8ドット標準のスプライトだけが使用できます。

SPRITE$
（スプライト
・ダラー）
（システム変数）

スプライトのパターンを決めることを「スプライトを定義する」と呼びます。

スプライトの定義は、ＳＰＲＩＴＥ$というシステム変数に値を入れることによって行います。

```
170  SPRITE$(J)=B$
```

スプライト番号（J）　文字列（B$）

・システム変数は、ＢＡＳＩＣが特別に使用する変数で、変数名を変えることができません。
・ＳＰＲＩＴＥ$は、文字型変数の１種ですが、表示したいパターンのドットを８ドットずつまとめて１つの文字とし、パターンの種類によって異なる数の文字を１つの文字列としてまとめたものを値にします。

例） ８×８ドット標準の場合

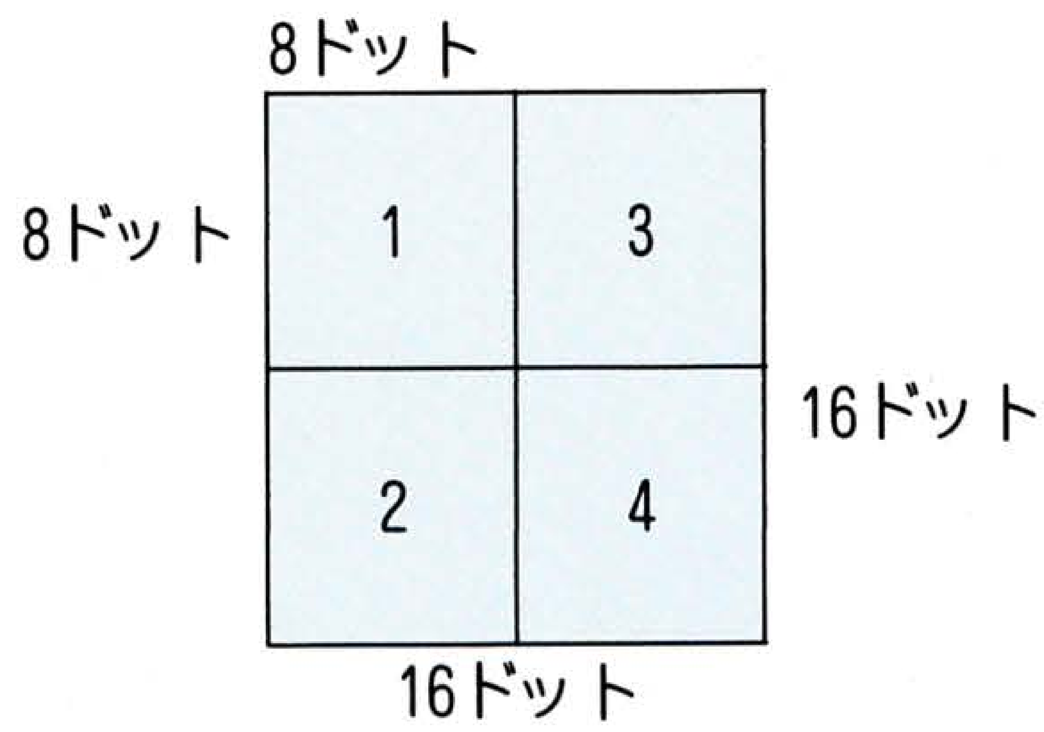
0	0	0	1	1	0	0	0	（&B00011000）=CHR$(24)
0	0	1	1	1	1	0	0	（&B00111100）=CHR$(60)
0	1	0	1	1	0	1	0	（&B01011010）=CHR$(90)
0	1	1	1	1	1	1	0	（&B01111110）=CHR$(126)
1	1	1	1	1	1	1	1	（&B11111111）=CHR$(255)
0	0	1	0	0	1	0	0	（&B00100100）=CHR$(36)
0	1	0	0	0	0	1	0	（&B01000010）=CHR$(66)
0	1	0	0	0	0	1	0	（&B01000010）=CHR$(66)

※SPRITE$(0)=CHR$(24)+CHR$(60)+CHR$(90)+CHR$(126)+CHR$(255)+C HR$(36)+CHR$(66)+CHR$(66)

※左の図の１の部分が、指定した色または前景色で画面に表示されます。

※&Bは２進数をあらわします。

・８×８ドット拡大のスプライトは、８×８ドット標準のスプライトと同じ様に定義します。画面に表示されるときには、定義されたパターンの１ドットが画面の４ドット（タテ２ドット×ヨコ２ドット）に拡大されます。
・16×16ドットのスプライトは、８×８ドットのパターン４個に分割し、次の順番で定義します。標準も拡大も定義の方法は同じです。



VAL
(バリュー)
(関数)

VALは、引数としてあたえられた文字列のあらわす数値を得る関数です。

数字でできた文字列を数値に変換するときに使います。

例)

```
A=VAL (B$)
C=VAL ("&HFF")
```

・文字列の先頭の文字が数字または+,−,&でないときは、VALは値として0に変換します
また、文字列の途中に数字以外の文字があるときは、そこから後ろの文字列は無視します。
ただし、スペース(空白)は無視して変換します。

・このプログラムでは、READ文でデータとして読んだ文字列を16進数に変換するため、文字列の先頭に"&H"を付けてVAL関数の引数にしています。

PUT
SPRITE
(プット
・スプライト)

PUT SPRITEはスプライトパターンを表示する命令です。

```
280 PUT SPRITE 1, (X, Y), 13, 4
```

1:スプライト面番号　(X, Y):位置座標　13:カラーコード　4:スプライトパターン番号

・**スプライト面番号**は、スプライトパターンを表示するスプライト面を指定する番号で、0〜31の数値または数値型変数で入力します。1つのスプライト面には1個のスプライトパターンだけを表示できます。2つ以上のスプライトパターンを同時に画面に表示するときは、それぞれ表示するスプライト面を変えてください。同じスプライト面に表示すると、先に表示したスプライトパターンが消えてしまいます。

・スプライト面番号は省略できます。省略したときは、最後に使用したスプライト面を指定したことになります。

・2つ以上のスプライトパターンが重なると、スプライト面番号の小さいスプライトパターンが番号の大きいスプライトパターンの上に重なって表示されます。

・スクリーン1〜3では、画面の水平方向に5個以上のスプライトパターンが並ぶと、スプライト面番号の小さいものから順に4個までは表示されますが、5個目からあとのスプライトパターンは消えてしまいます。スクリーン4〜12では、8個まで並べて表示できます。この理由で消えたスプライトパターンは、並んでいるスプライトパターンの数が減れば再び表示されます

・**位置座標**は、スプライトパターンを表示する位置を指定する数値です。
指定した位置がスプライトパターンの左上角になります。ヨコ座標は-32〜255,タテ座標は-32〜191(スクリーン5〜8では-32〜211)の範囲で指定します。
スクリーン6と7では、絵や図形を表示するときのヨコ座標の範囲が511までになりますが、スプライトの座標は255までを使用します。(画面右端が255です。)スプライトパターンが表示される大きさも変わりません。

・位置座標は省略できます。省略したときは、そのスプライト面に最後に表示したスプライトパターンの位置を指定したことになります。

PUT SPRITE のつづき

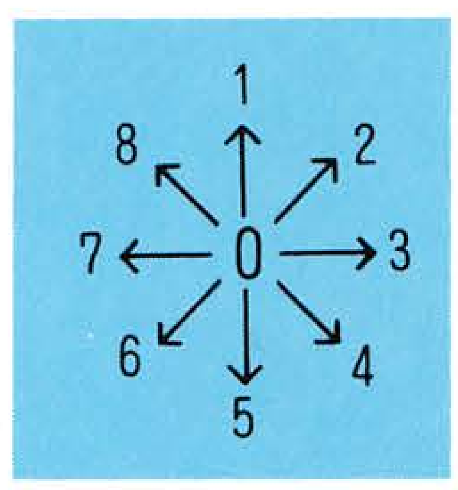
・タテ座標に 208（スクリーン4～8では 216）を指定すると、その面からあとのすべてのスプライト面の表示が消えます。
・タテ座標に 209（スクリーン4～8では 217）を指定すると、そのスプライト面の表示が消えます。
・**カラーコード**は、スプライトパターンを表示する色を番号で指定します。
指定できるカラーコードの範囲は、画面モードによって変わります。(⇨62ページ)
省略したときは、前景色を指定したことになります。
・**スプライトパターン番号**は、ＳＰＲＩＴＥ$変数を使って定義したスプライトパターンを番号で指定します。定義していない番号を指定すると、画面に意味のない模様がスプライトとして表示されることがあります。
・スプライトパターン番号は省略できます。省略したときは、そのスプライト面に最後に表示したスプライトパターンを指定したことになります。

ＳＴＩＣＫ（スティック）（関数）

ＳＴＩＣＫは、カーソルキーやジョイスティックまたはジョイパッドがどの方向に押されているかを調べる関数です。

```
320  S=STICK (0)
```
（0）……機器の番号

・調べた結果は、数値で得ます。



・**機器の番号**は、次のように数値で指定します。
0 ………カーソルキーを調べるとき
[↑] が押されていれば1、[→]が押されていれば3、[↑]と[→]がいっしょに押されていれば2になる
1 ………汎用ポート1に接続したジョイスティックまたはジョイパッドを調べるとき
2 ………汎用ポート2に接続したジョイスティックまたはジョイパッドを調べるとき

ON GOTO
(オン・
ゴートゥー)

ＯＮ　ＧＯＴＯは、指定された変数の値によって次に実行するプログラムの行を変える命令です。

```
380 ON T+1 GOTO 400,430
```

T+1：変数名　400：行番号　430：行番号

指定された変数の値が１なら左から１番目の行番号へ、２なら２番目の行番号の行へというようにジャンプします。

・**変数名**の部分は、数値や計算式でもかまいません。
・変数の値が１より小さいときや指定された行番号の数より大きいときは、何もしないで次の行に進みます。
・変数の値が整数でないときは、小数部を無視します。

このプログラムでは、状態（右向きか左向きか）をあらわす変数の値により、次にスプライトパターンを表示する位置を決める行を変えています。

《テクニック》

8×8ドットのスプライトパターンの定義方法

8×8ドットのスプライトパターンを定義するときのテクニックとして、下の方法をおすすめします。

```
100 FOR N=1 TO 8
110 READ A$
120 B$=B$+CHR$(VAL("&B"+A$))
130 NEXT N
140 SPRITE$(C)=B$
          :
900 DATA 00011000
910 DATA 00111100
920 DATA 01011010
930 DATA 01111110
940 DATA 11111111
950 DATA 00100100
960 DATA 01000010
970 DATA 01000010
```

・ＤＡＴＡ文を０と１だけで表記する（900行～）
・読み取ったデータを２進数として数値に変換してから文字列に変換する（120行）

この方法なら、ＤＡＴＡ文を入力するときにスプライトパターンを目で確かめながら作ることができます。

第4章

音楽がない生活なんて考えられない！という人のパソコンライフのために

BASICでMUSIC!

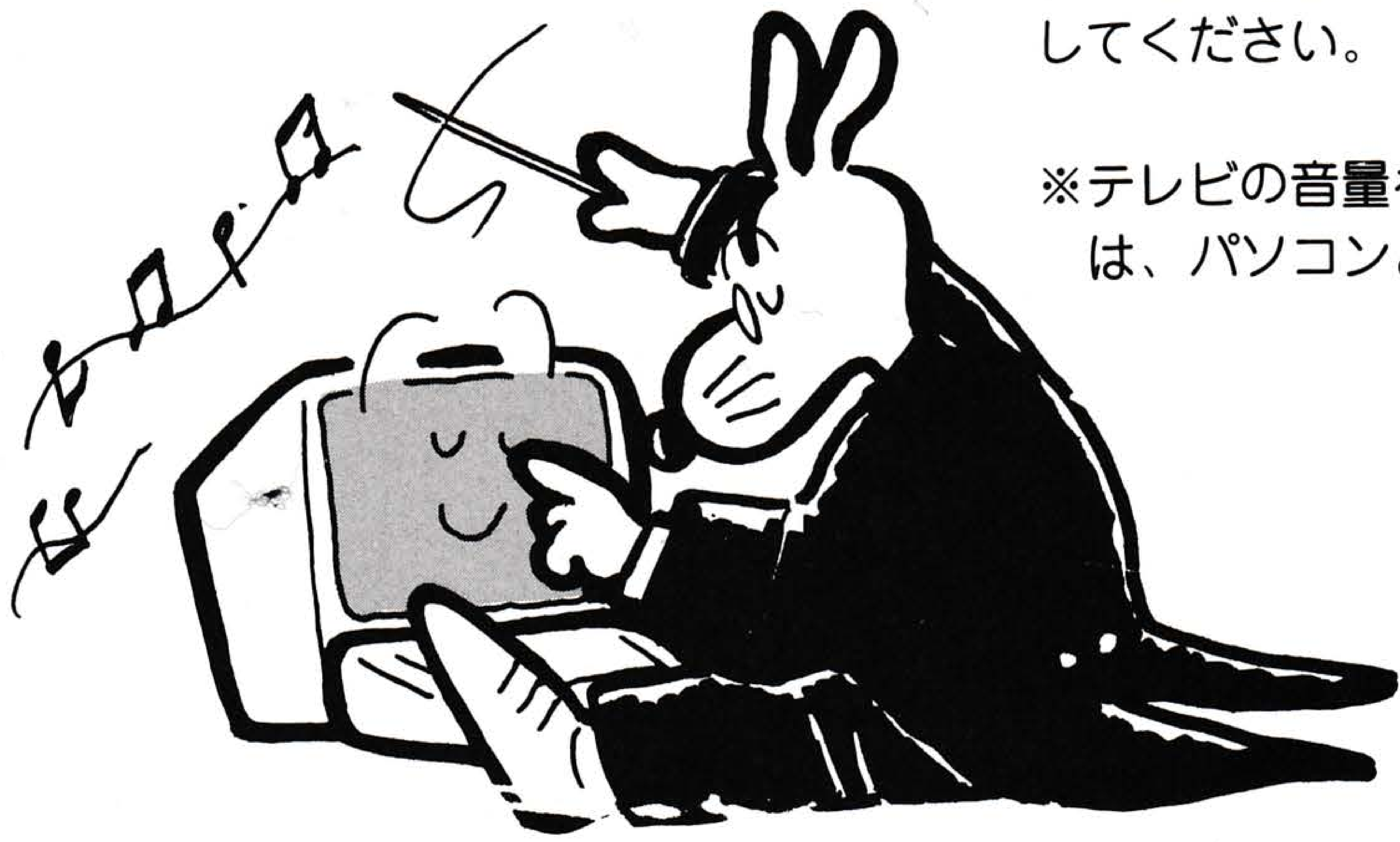


まず♪ドレミから

パソコンでゲームをしたことのある人は、ゲームがスタートするときなどにきれいな音楽が流れるのを聞いたことがあるでしょう。あなたのMSX２＋パソコンにも、音楽を演奏する機能がついています。ＢＡＳＩＣで音楽を演奏させてみましょう。

演奏が終わればプログラムは止まりますが、途中で止めたいときは CTRL を押しながら STOP を押してください。

※テレビの音量を大きくしても何も聞こえないときは、パソコンとテレビの接続を確かめてください。



リストと解説

```
1 'save"ﾄﾞﾚﾐ.bas"
10 FOR N=0 TO 1
20   PLAY "V15O4CDEFGABO5C8.R
16"
30   PLAY "CO4BAGFEDC8.R16"
40 NEXT N
```

“ドレミファソラシド”を繰り返し演奏するプログラムです。

```
1 'save"ｷﾘﾂﾚｲ.bas"
10 PLAY"T90V15O4L2","T90V15O4
L2","T90V15O4L2"
20 PLAY "EDE","GGG","O5CO4BO5
C"
```

学校で聞いた覚えがあるでしょう？
和音を鳴らすプログラムです。

```
1 'save"ｴﾘｰｾﾞ.BAS"
10 PLAY "V15T120L8O5EE-EE-EO4
BO5DCO4A4."
20 PLAY "R8CEAB4.R8EG#BO5C4.R
8"
```

半音を使った演奏のプログラムです。

新しい命令の説明

MSX2+パソコンの音楽機能

MSX 2+パソコンでは、音楽の演奏などの音は「PSG」（ピー・エス・ジー）というLSIを使って発生しています。

PSGは、発生する音の高さや大きさを調整する回路を3組もっており、同時に3つの音を発生することができます。そこでBASICでは、この3組の回路を「ボイスチャンネル1～3」として、別々に操作できるようになっています。

PLAY（プレイ）

音楽を演奏したり、演奏について各種の設定を行う命令です。

```
20 PLAY "EDE", "GGG","O5CO4BO5C"
        "文字列"  "文字列"   "文字列"
```

・″文字列″は、文字型変数や文字列の連結式でもかまいません。
・″文字列″は、**「MML」**（ミュージック・マクロ・ランゲージ）を値にします。
MMLは、一定の文字や数字・記号の集まりで、音楽の演奏についての各種の細かい命令の役割をします。MMLについては、182ページの一覧表にのっています。
・1つのPLAY文で、3つまでの″文字列″を入力することができます。
左側の″文字列″から順にボイスチャンネル1、ボイスチャンネル2‥‥というように、各ボイスチャンネルについての命令とみなされます。したがって、ボイスチャンネル2や3で演奏させるときは、ボイスチャンネル1を使わないときでも、前に「″″」（ヌル：値がない文字列）を入力することが必要です。
例）　PLAY "", "", "CDE"
・1つのPLAY文に2つ以上の″文字列″が入力されているときは、各文字列に書かれたMMLが同時に実行（演奏）されます。
例）　PLAY "C", "E", "G"　　ドミソの和音になります。
　　　PLAY"CC", "EF", "GA"　　ドミソとドファラの2和音を演奏します。

サンプルプログラムで使用したMML

・O…………そのあと演奏する音のオクターブの高さを後ろの数字で指定します。
数字の範囲は1～8で、スタート時は4になっています。この指定は、次に「O」を指定するときまで有効です。つまり、プログラムの実行を終わっても設定が残ったままですので、そのあとで別のプログラムを実行したときにも影響します。

・L…………そのあと演奏する音の長さを後ろの数字で指定します。
数字の範囲は1～64で、「何分音符」かを指定します。スタート時は4です。
この指定は、次にLを指定するまで有効です。

・A～G……発生する音の高さをそのオクターブの中での音程で指定します。
C（ド）D（レ）E（ミ）F（ファ）G（ソ）A（ラ）B（シ）
後ろに数字を付けると、直前の音の長さを指定したことになります。

・「.」……付点。直前の音の長さを1.5倍します。

・R…………休符。後ろの数字で「何分休符」かを指定します。
休符はLによる音の長さの指定に影響されないので注意してください。
数字を付けないときは4分休符になります。

・T…………テンポの指定。後ろの数字で1分間に4分音符で何拍するテンポかを指定します。
数字の範囲は32～255で、スタート時は120です。
この指定は、次にTを指定するまで有効です。

・V…………音量。後ろの数字で音の大きさを指定します。数字の範囲は0～15で、スタート時は8です。この指定は、次にVを指定するまで有効です。

・－…………♭（フラット）。直前の音の高さを半音下げます。

・#…………シャープ。直前の音の高さを半音上げます。＋を代用できます。

MML使用時の注意

・各"文字列"のMMLは、そのボイスチャンネルだけに有効です。たとえば、2つ以上のボイスチャンネルを使って演奏する曲のテンポを変えるときには、同じPLAY文の中のすべての"文字列"でTを指定し直すことが必要です。

・O，L，T，Vの指定は、そのボイスチャンネルで次に指定し直すまで有効です。
したがって、これらの設定を変更するプログラムを実行させたあとで別の演奏プログラムを実行させるときは、最初にこれらを指定し直してください。

・Oは数字の0と特にまちがいやすいので注意して入力してください。

参考プログラム

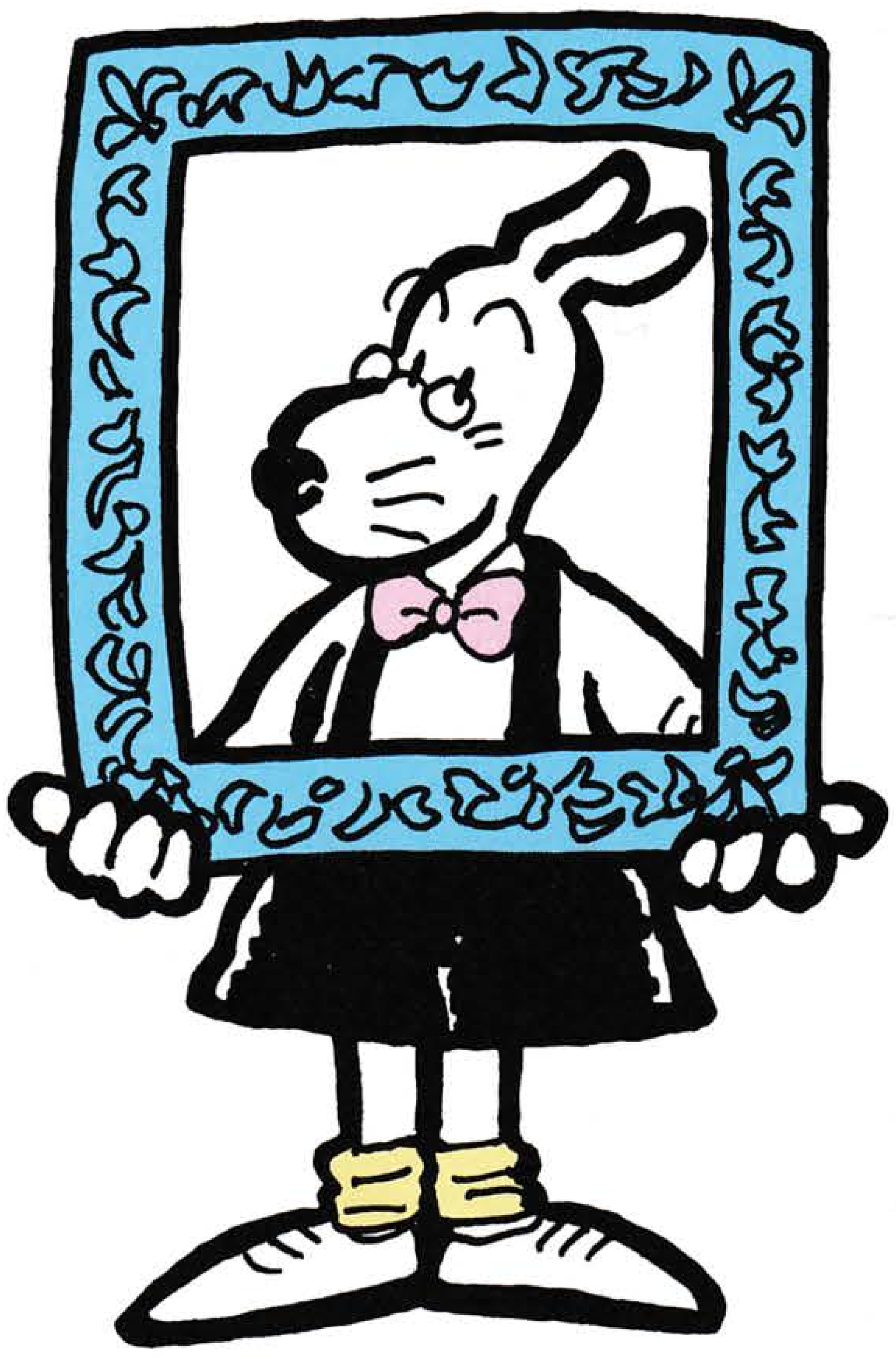
```
1 'save"バッハ.BAS"
100 PLAY "T160L6V15","T160L2V
12"
110 PLAY "R6O6GABO7DCCED","O4
GO5GE"
120 PLAY "DGF#GDO6BGAB","O4BO
5EO4E"
130 PLAY "O7CDEDCO6ABG","ABO5
C"
140 PLAY "F#GADF#AO7CO6BA","O
5DF#D"
150 PLAY "BGABO7DCCED","GGC"
160 PLAY "DGF#GDO6BGAB","O4BO
5ED"
170 PLAY "EO7DCO6BAGDGF#","CC
#D"
180 PLAY "G2","G"
```

```
1 'save"ハナ.bas"
10 PLAY "T60V15L16","T60V12L1
6"
20 PLAY "O5D8.DG8.GAGF#ED4","
O5D8.DO4B8.BCO5EDO4CB4"
30 PLAY "O4BO5DGAB8O6D8O5A4.R
8","O4B8BO5DG8G8F#4.R8"
40 PLAY "O5D8.DG8.GAGF#ED4","
O5D8.DO4B8.BCO5EDO4CB4"
50 PLAY "O5A8.AA8B8G4.R8","O4
A8.O5F#F#EDCO4B4.R8"
```

２本とも２パートの同時演奏を行うプログラムです。どちらも今まで説明したことだけを使って作られていますので、演奏プログラムを作るときの参考にしてください。

「展覧会の絵」を演奏しよう

ムソルグスキーの「展覧会の絵」の一部を演奏するプログラムです。



このプログラムは演奏が終わると止まります。

※途中で止めたいときは、CTRL を押しながら STOP を押してください。

リストと解説

```
1 'save"テンランカイ.bas"
100 READ A$,B$,C$:IF A$="END"
 THEN END
110 PLAY A$,B$,C$
120 GOTO 100
130 '* えんそう データ *
140 DATA V13L4,V10L4,V10L4
150 '
160 DATA L4O4GFB-O5C8F8D,,
170 '
180 DATA C8F8DO4B-O5CO4GF,,
190 '
200 DATA O4G FB-O5C8F8D
210 DATA O4D CD   A   A
220 DATA O3B-AB-O4C   F
230 '
240 DATA O4C8F8DO4B-O5CO4GF
```

- 100: 140行からのＤＡＴＡ文からデータを読み、もしデータが「END」なら、実行を終了する
- 110: 読み取ったデータにしたがって演奏する
- 120: 100行に戻る
- 140: 各ボイスチャンネルの音量と音の長さのデータ
- 160〜180: ボイスチャンネル１だけの演奏データ（他のチャンネルはまだ演奏しない）
- 200〜220: ３チャンネルで演奏するデータ（３行で１組み）
  ※ＭＭＬのあいだの空白は、リストを見やすくするために挿入したもので、演奏時には無視されます。

```
250 DATA O4A    B- G    G   CC
260 DATA O4C    C   D    EO3GA
270 '
280 DATA O4B-O5C8F8DC8F8DO4B-

290 DATA O4F    A8R8B-A8R8B- F

300 DATA O3D    C  O2B-A    GO3G


310 '
320 DATA O5CO4GF   G FB-
330 DATA O4G   EC   D CD
340 DATA O3C   EFO2B-AG
350 '
360 DATA O5C8F8D O4B-O5E -CO4
B-
370 DATA O4A8R8B-   G O5CO4A
F
380 DATA O2F    B-O3G    C   F
B-
390 '
400 DATA END,,
```

新しい命令の説明

このプログラムでは、特に新しい命令を使っていませんが、MMLの組み合わせかたを理解するための参考にしてください。

キーボードで音楽を

パソコンのキーボードを使って音楽を演奏できるようにするプログラムです。
これなら、演奏プログラムを作るのがメンドウだという人もＯＫですね？

※プログラムを止めたいときは、[CTRL]を押しながら[STOP]を押してください。



リストと解説

```
1 'save"ｷｰﾎﾞｰﾄﾞ.bas"
100 SCREEN 1:WIDTH 32:CLS
110 PLAY "T120":VV=15:LL=8:GO
SUB 410:GOSUB 430
120 '
130 PLAY "XA$;":A$=""
140 '
150 K$=INPUT$(1)
160 IF K$=CHR$(&H1E) AND  VV
<15 THEN VV=VV+1:GOSUB 410
170 IF K$=CHR$(&H1F) AND 0<VV
    THEN VV=VV-1:GOSUB 410
180 IF K$=CHR$(&H1C) AND 1<LL
    THEN LL=LL/2:GOSUB 430
190 IF K$=CHR$(&H1D) AND   LL
<64 THEN LL=LL*2:GOSUB 430
200 IF PLAY(0) THEN 140
```

- 100：画面の初期設定
- 110：演奏のテンポと音量と音の長さを指定する変数の値を初期設定し、 410行から始まるサブルーチン、430行から始まるサブルーチンにジャンプする
- 130：A$の値のMMLにより音を鳴らす
- 150：押されているキーを調べる
- 160：[△]が押されていて、変数ＶＶの値が15未満なら、ＶＶの値を１増やして410行へ
- 170：[▽]が押されていて、変数ＶＶの値が０より大きければ、ＶＶの値を１減らして410行へ
- 180：[▷]が押されていて、変数ＬＬの値が１より大きければ、ＬＬの値を半分にして430行へ
- 190：[◁]が押されていて、変数ＬＬの値が64未満なら、ＬＬの値を２倍にして430行へ
- 200：音が鳴っている途中なら、140行へ

```
210 IF K$="z" OR K$="Z" THEN
A$="O4A"
220 IF K$="x" OR K$="X" THEN
A$="O4B"
230 IF K$="c" OR K$="C" THEN
A$="O5C"
240 IF K$="v" OR K$="V" THEN
A$="O5D"
250 IF K$="b" OR K$="B" THEN
A$="O5E"
260 IF K$="n" OR K$="N" THEN
A$="O5F"
270 IF K$="m" OR K$="M" THEN
A$="O5G"
280 IF K$="," THEN A$="O5A"
290 IF K$="." THEN A$="O5B"
300 IF K$="/" THEN A$="O6C"
310 IF K$="a" OR K$="A" THEN
A$="O4G#"
320 IF K$="s" OR K$="S" THEN
A$="O4B-"
330 IF K$="f" OR K$="F" THEN
A$="O5C#"
340 IF K$="g" OR K$="G" THEN
A$="O5E-"
350 IF K$="j" OR K$="J" THEN
A$="O5F#"
360 IF K$="k" OR K$="K" THEN
A$="O5G#"
370 IF K$="l" OR K$="L" THEN
A$="O5B-"
380 IF K$=":" THEN A$="O6C#"
390 IF K$="]" THEN A$="O6E-"
400 GOTO 120
```

キーボードの下から１段目のキーにピアノの白鍵にあたる音をそれぞれ割り当て、押されたキーの高さの音をＡ$に入れる

２段目のキーに黒鍵の音を割り当てる

120行に戻る

```
410 '
420 PLAY "V=VV;":LOCATE  5,10
:PRINT "V=";VV:RETURN
430 '
440 PLAY "L=LL;":LOCATE 15,10
:PRINT "L=";LL:RETURN
```

（以下はサブルーチン）

420: 音量を変数ＶＶの値に設定し、画面に表示してメインルーチンに復帰する

440: 音の長さを変数ＬＬの値に設定し、画面に表示してメインルーチンに復帰する

新しい命令の説明

PLAY（関数）

指定されたボイスチャンネルが演奏中かどうかを調べる関数です。

```
A=PLAY (0)
```
変数名　チャンネル番号

・チャンネル番号は0～3の数値で指定します。各数値は下の意味になります。

0……ボイスチャンネル1～3のどれか
1……ボイスチャンネル1
2……ボイスチャンネル2
3……ボイスチャンネル3

・調べた結果は、演奏中ならー1、演奏中でなければ0になります。
・ここでいう「演奏中」とは、ＰＬＡＹ文を実行中かどうかを意味します。
したがって、休符や各種の設定を実行中で実際には音が鳴っていなくても「演奏中」になります。

新しく出てきたMML

＝変数名；……MMLの後ろに付ける数字を変数で指定するときに使います。

```
420 PLAY "V=VV;"
```
MML　変数名

・変数名は数値型変数であることが必要です。

X変数名；……入力するMMLを変数で指定するときに使います。

```
130 PLAY "XA$;"
```
変数名

・変数名は文字型変数であることが必要です。

効果音を作ろう

ＢＡＳＩＣの音楽機能は、音を鳴らすだけでなく音色そのものを作ることができます。これを使って、怪獣出現のシーンをプログラムしてみました。



リストと解説

```
1 'save"コウカオン.bas"
100 '* けいほう *
110 SOUND 7,60             PSGのレジスタ7にチャンネルを指定
120 SOUND 0,10             チャンネルAの周波数を設定
130 SOUND 1, 5
140 SOUND 2, 0             チャンネルBの周波数を設定
150 SOUND 3, 5
160 SOUND 8,15             音量を設定する
170 FOR N=1 TO 1000:NEXT N 一定時間鳴らし続ける
180 '* かいじゅう *
190 SOUND  7,54            レジスタ7にノイズの使用を設定
200 SOUND  0, 0
210 SOUND  1,11
220 SOUND  8,16            音量をエンベロープモードに設定
230 SOUND  6,18            ノイズ周波数の設定
240 SOUND 11, 0            エンベロープ周期の設定
250 SOUND 12,30
260 SOUND 13, 8            エンベロープ形状の設定
270 FOR N=1 TO 3000:NEXT N
280 SOUND  8, 0            音量を0にする
```

新しい命令の説明

SOUND
(サウンド)

PSG（MSX2+パソコンの音楽機能を受け持つLSI）を直接操作する命令です。

```
110 SOUND 7, 60
```
レジスタ番号　設定数値

PSGは下の表のように0～13までのレジスタ（各種の設定を記憶するメモリ）をもっています。SOUND文は、このレジスタに直接値を書き込みます。

レジスタ番号	機　能								
0	チャンネルA周波数	FT A　（8ビット）　0～255の数値で設定							
1		CT A　（4ビット）　0～ 15の数値で設定							
2	チャンネルB周波数	FT B　（8ビット）　0～255の数値で設定							
3		CT B　（4ビット）　0～ 15の数値で設定							
4	チャンネルC周波数	FT C　（8ビット）　0～255の数値で設定							
5		CT C　（4ビット）　0～ 15の数値で設定							
6	ノイズ周波数	NP　（5ビット）　0～ 32の数値で設定							
7	チャンネル 設定	bit7	bit 6	bit5	bit4	bit3	bit2	bit1	bit0
		IN OUT		ノイズ			トーン		
		IOA	IOB	C	B	A	C	B	A
8	チャンネルA音量	(4ビット+エンベロープモードのサインビット1ビット) 0～15のとき音量固定。16以上はエンベロープモード。							
9	チャンネルB音量	(4ビット+エンベロープモードのサインビット1ビット) 0～15のとき音量固定。16以上はエンベロープモード。							
10	チャンネルC音量	(4ビット+エンベロープモードのサインビット1ビット) 0～15のとき音量固定。16以上はエンベロープモード。							
11	エンベロープ周期	FT　（8ビット）　0～255の数値で設定							
12		CT　（8ビット）　0～255の数値で設定							
13	エンベロープ形状	0～15の数値で設定							

音色の作成は、ある程度、経験的に行うものです。とにかく設定して鳴らしてみてください。
各レジスタに設定する値の計算方法については、市販の解説書などを参照してください。

「もっと音楽を!」という人は…

ＭＳＸ２+ パソコンの音楽機能を受け持つＰＳＧは、多くのパソコンで使用されている定評のある音源ＬＳＩです。

しかし、パソコンを使った音楽演奏に特に興味のある人にとっては、少し物足りないかも知れません。
「もっとリアルな音色で演奏させたい！」とか「もっと簡単に音色を変更したい！」とか感じた人のために、ＦＭ音源用拡張ＢＡＳＩＣがあることをお知らせしておきます。

ＦＭ音源用拡張ＢＡＳＩＣ　（ＭＳＸ－ＭＵＳＩＣ）

流行のＦＭ音源を内蔵する機種のＭＳＸ２+ パソコンやＦＭ音源カートリッジのために作られた拡張ＢＡＳＩＣです。

・あらかじめ用意されている62種類のリアルな音色と５種類のリズム音を使って、最大11種類の音を同時に使って演奏させることができます。
・音楽を演奏中でも他の命令を実行するＢＧＭ機能があり、ゲーム中の演奏に便利です。
・音色を番号で指定する「ＶＯＩＣＥ」（ボイス）や 200分の１音単位で音の高さを調節できる「ＴＲＡＮＳＰＯＳＥ」（トランスポーズ）などの強力な命令をもっています。
・ＰＳＧ用とほとんど共通のＭＭＬを使っているため、ＰＳＧ用に作った演奏プログラムの移植が容易です。

第5章

漢字を使いにくいパソコンなんて・・・という人のための正しい解決方法

かんたん漢字モード!

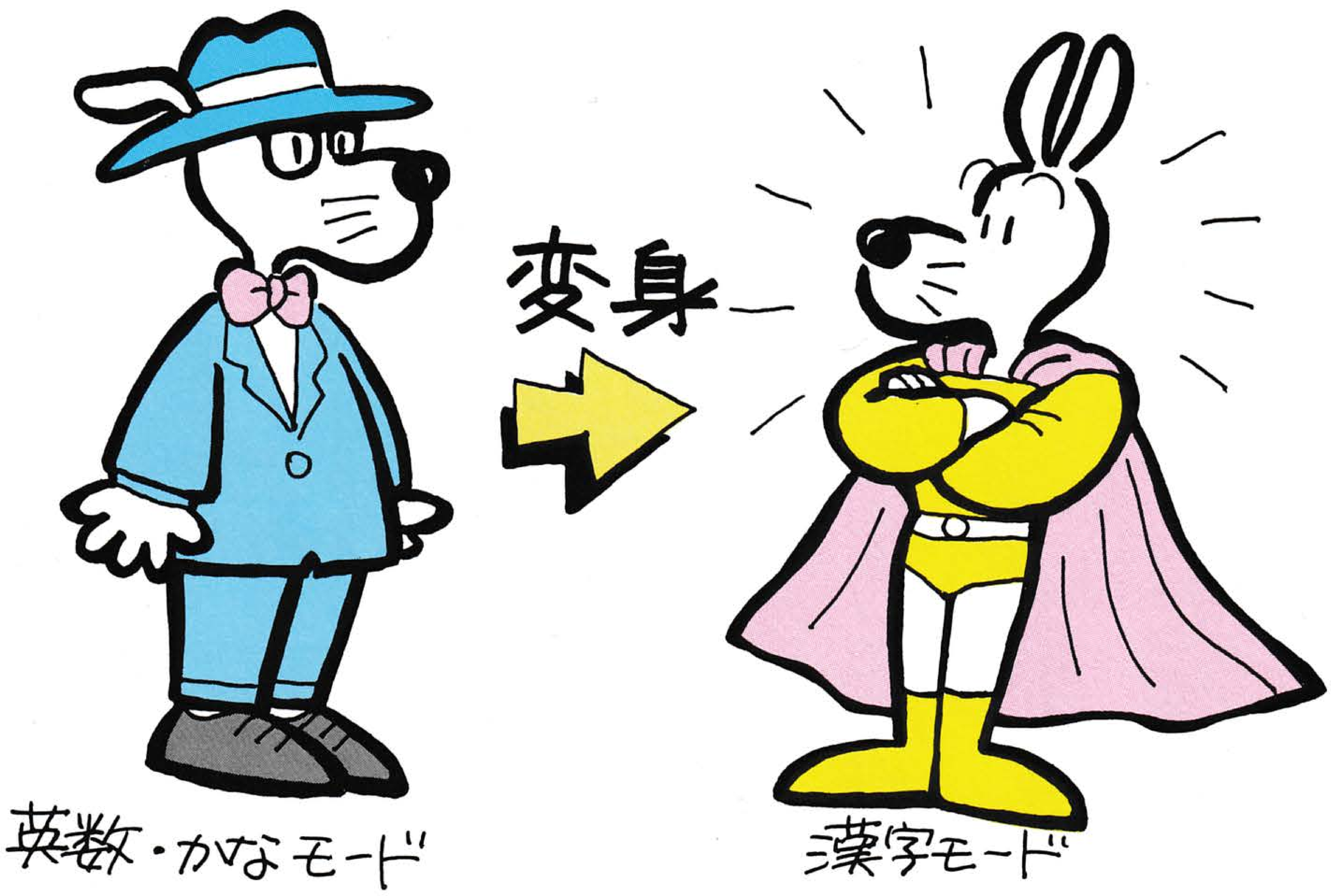


漢字モードのスタート

これからのページでは、ＭＳＸ２+ 漢字ＢＡＳＩＣの特長のひとつである漢字モードについて説明します。

漢字モードって何?

ＭＳＸ２+ 漢字ＢＡＳＩＣでは、これまでのＭＳＸやＭＳＸ２ＢＡＳＩＣにくらべ、ずっとかんたんに漢字が使えるようになっています。ただ、これまでのＢＡＳＩＣと同じやりかたでも漢字をあつかえるようにしているため、電源スイッチを入れたときから新しい漢字機能を使うことはできません。まずＢＡＳＩＣに「頭の切り替え」をさせることが必要なのです。
この「頭の切り替え」をした状態を**「漢字モード」**と呼びます。



それに対して、これまで説明してきたＢＡＳＩＣがスタートしたときのままの状態を**「英数・かなモード」**と呼びます。

漢字モードの特長

・キーボードから「かな」で読みを入力し、画面上で変換することにより、漢字を直接入力できる
・プログラムの中でＰＲＩＮＴ文やＩＮＰＵＴ文の文字列（文字定数）やＲＥＭ文に漢字を使える
・漢字の文字列を操作するための新しい命令が使える

これらが漢字モードの主な特長です。

漢字モードのスタートのしかたと終了のしかた

```
CALL KANJI
```

ＢＡＳＩＣを英数・かなモードから漢字モードに切り替えるには、上の命令を実行させせます。

ダイレクトモードでは画面の表示が消え、左上にそれまでより大きな文字で「Ｏｋ」と表示されます。これでＢＡＳＩＣが漢字モードに切り替わっています。

この命令を**「ＣＡＬＬ　ＫＡＮＪＩ（コール・カンジ）命令」**と呼びます。

```
CALL ANK
```

漢字モードから英数・かなモードに切り替えるには、上の命令を実行させせます。

ダイレクトモードでは再び画面の表示が消え、左上に元の大きさの文字で「Ｏｋ」と表示されます。これで英数・かなモードに切り替わっています。

この命令を**「ＣＡＬＬ　ＡＮＫ（コール・アンク）命令」**と呼びます。

漢字モードでの命令などの入力のしかた

まずＢＡＳＩＣを漢字モードに切り替え、英数・かなモードと同じように、キーボードの文字キーを押してみましょう。

英数・かなモードと同じように、押したキーの文字が画面に表示されますね。

ただ、文字のタテの大きさが英数・かなモードより大きくなりますので、文字によってはかなり形が違って見えます。まちがえないように注意してください。

この大きさの文字を**「半角文字」**と呼びます。半角文字は英文字、数字、記号、ひらがな、カタカナ、グラフィック文字です。（これに対して、漢字や漢字と同じ大きさで表示される英文字、数字、記号、ひらがな、カタカナを**「全角文字」**と呼びます）

ご注意：ＢＡＳＩＣの命令や行番号、数値、変数名、演算子、区切り記号、ファイルスペックなどは、半角文字で入力してください。

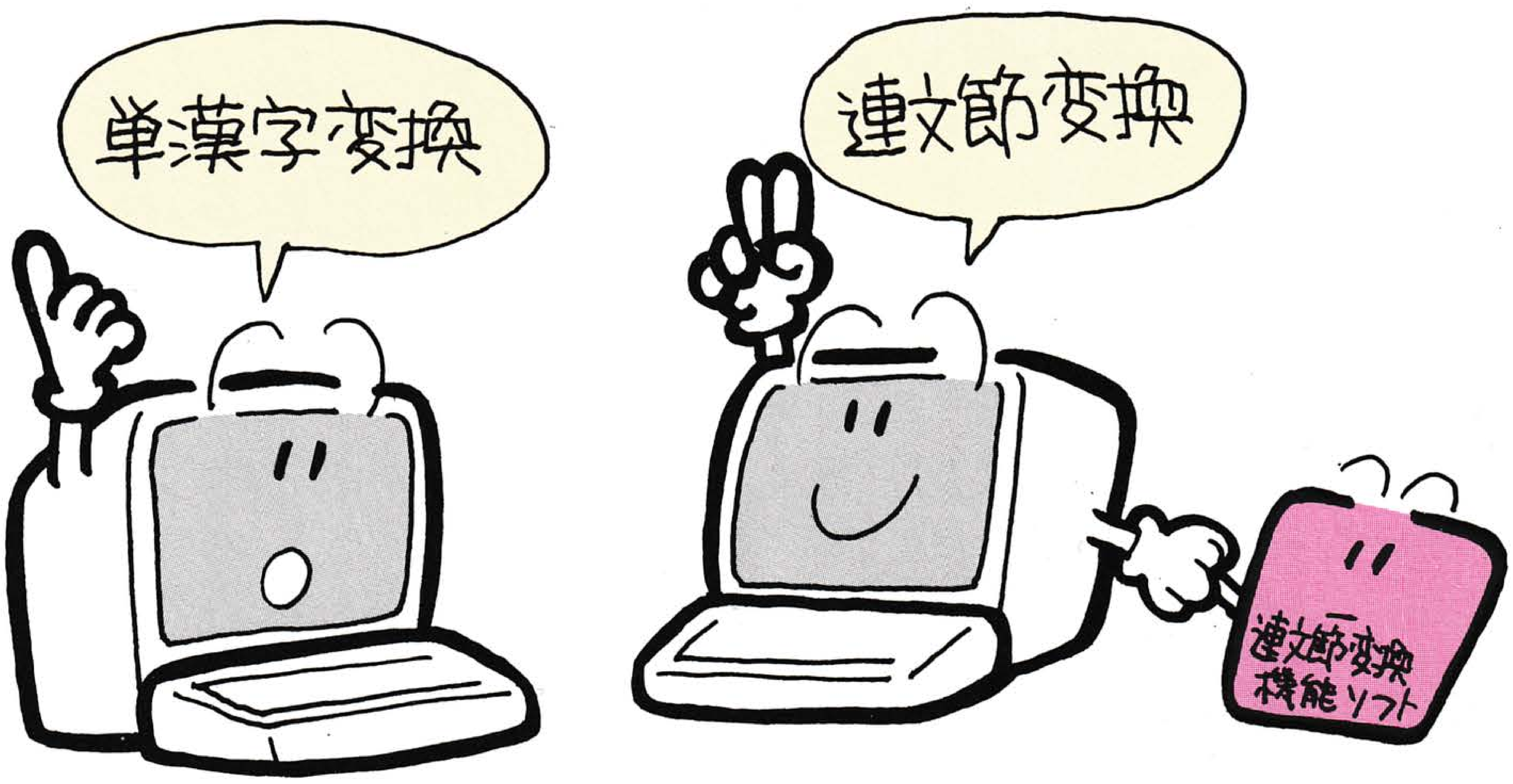
漢字の入力のしかた

漢字などの全角文字を入力する2つの方法

漢字などの全角文字を入力する場合は、まず CTRL を押しながら SPACE を押します。
（GRAPH を押しながら SELECT を押しても同じことができます）
これでＢＡＳＩＣは、キーボードから入力された文字を漢字（全角文字）の読みとして受け取る状態になります。
このあとの操作は、ＭＳＸ２＋パソコンの機種によって下の２通りに分かれます。

連文節変換機能をもつ機種　　**単漢字変換機能をもつ機種**

「連文節変換機能」とは、いくつかの「文節」（文を単語にならない程度に最も短く区切った単位）を一度に入力し、一度に漢字に変換する機能のことです。これはＭＳＸ２＋ＢＡＳＩＣのもつ機能ではなく、別に連文節変換用ソフトを内蔵しているパソコンだけがもつ機能です。
「単漢字変換機能」とは、漢字一文字分ずつの読みを入力しては漢字に変換する機能のことです。こちらは、ＭＳＸ２＋ＢＡＳＩＣが標準でもっている機能です。



・連文節変換機能をもつＭＳＸ２＋パソコンでは、ＢＡＳＩＣを漢字モードに切り替えたときは、連文節変換で漢字を入力するように自動的に設定されます。
・連文節変換機能を内蔵していないＭＳＸ２＋パソコンでも、連文節変換用ソフトを組み込んだワープロソフトのカートリッジなどを接続している場合は、ＢＡＳＩＣを漢字モードに切り替えたときに、自動的に連文節変換で漢字を入力するようになります。

以後の説明では、まず連文節変換で漢字やその他の文字を入力する方法を説明し、そのあとで単漢字変換で漢字などの文字を入力する方法を説明します。

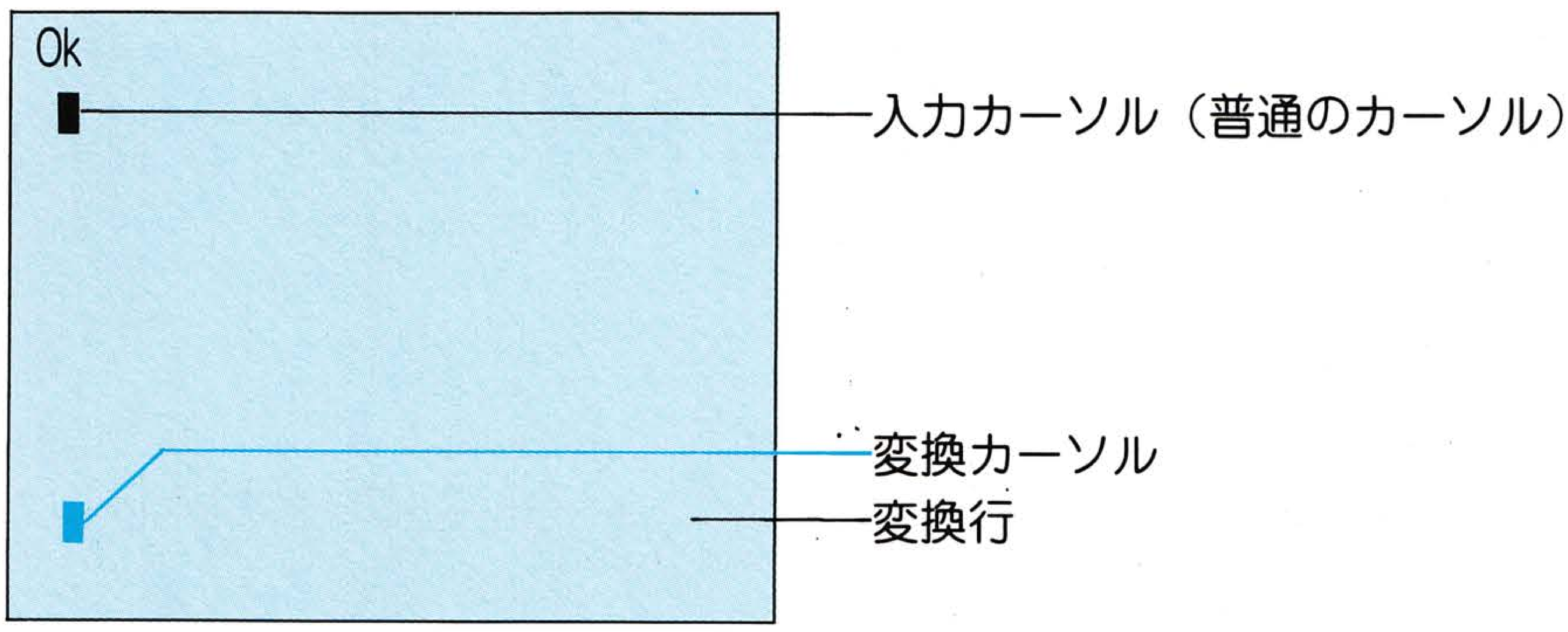
※この本で「連文節変換用ソフト」と呼んでいるのは、「ＭＳＸ－ＪＥ」（アスキー製）」のことです。ＭＳＸ－ＪＥは、「ＭＳＸ標準日本語入力フロントエンドプロセッサ」とも呼ばれ、各種のＭＳＸ用ソフトに組み込まれています。

連文節変換による文章の入力のしかた

連文節変換は、連文節変換用ソフトを内蔵している機種、または連文節変換用ソフトが組み込まれたワープロなどのカートリッジを接続しているＭＳＸ２+ パソコンでないとできません。それ以外のＭＳＸ２+ パソコンでは、単漢字変換で漢字を入力してください。（⇨107ページ）

連文節変換の画面

CTRL を押しながら SPACE を押したとき、画面の一番下の行にもうひとつカーソルが表示されます。これを**「変換カーソル」**と**「変換行」**と呼びます。

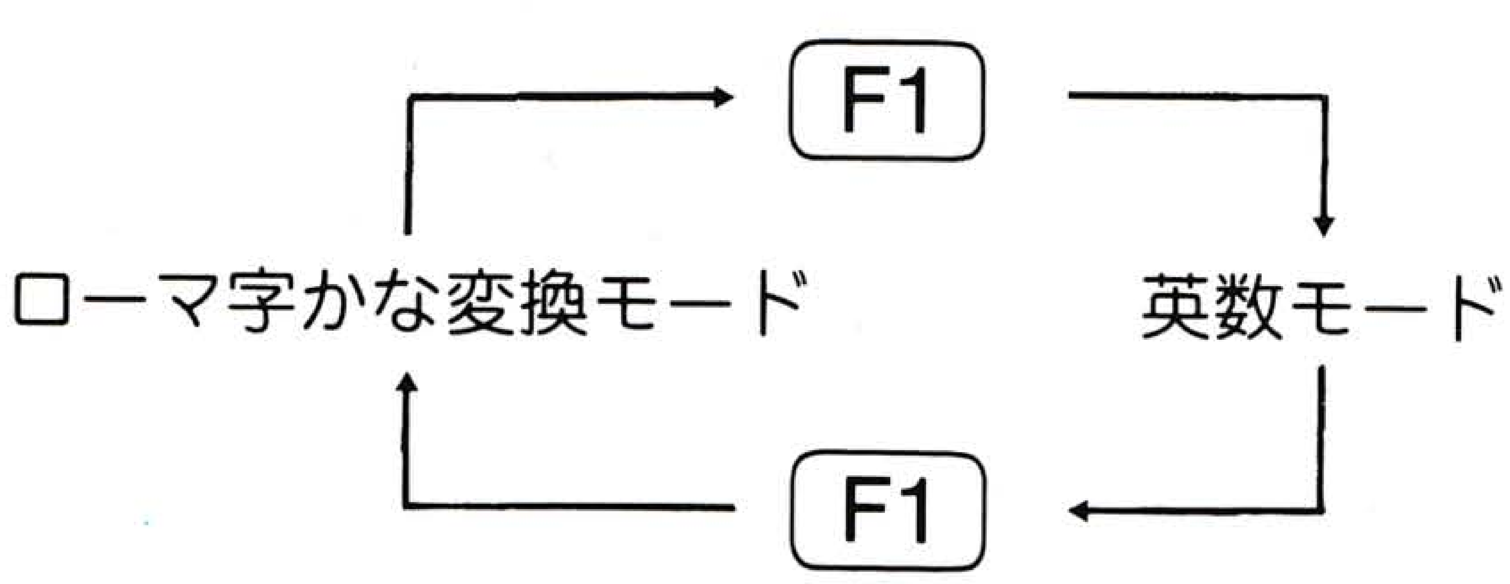


もう一度 CTRL を押しながら SPACE を押すと、変換カーソルが消え、漢字の入力ができなくなります。

連文節変換の２つの入力モード

連文節変換には**「ローマ字かな変換モード」**と**「英数モード」**の２つのモードがあります。

・変換カーソルが表示された直後は、ローマ字かな変換モードになっています。
・ローマ字かな変換モードと英数モードは、 F1 （入力モード切替）を押すたびに切り替わります。



ローマ字かな変換モードでは、次のようにして変換行に文字を入力します。

・かなランプが消えているとき………… [B] [O] ⇨ **ぼ**
・かなランプが点灯しているとき……… [ほ] [゛] ⇨ **ぼ**
・カタカナは直接には変換行に入力できません。一度ひらがなで入力してから、 [F2] （カタカナ変換）を押してカタカナに変換します。
・数字はそのままキーを押して入力できます。
・記号のキーを押すと、かな記号が入力されます。かな記号がないキーでは、英記号が入力されます。
・英文字は原則として入力できません。英文字を入力したいときは、英数モードに切り替えてください。

英数モードで文字キーを押すと、次のように変換行に入力されます。

・かなランプが消えているときには、英文字、英記号、数字が入力されます。
・かなランプが点灯しているときには、ひらがな、かな記号が入力されます。

これ以後は、ローマ字かな変換モードで操作しているものとして説明します。

連文節変換モードでの各キーの機能

パソコンの機種によっては、連文節変換ができるワープロソフトを内蔵しているものもありますが、ＢＡＳＩＣの漢字モードでの連文節変換に使用するキーや、各キーの機能は同じではありませんから注意してください。

次ページに、ＢＡＳＩＣの漢字モードでの連文節変換で使用できる特殊キーの一覧表を掲げています。

キーの名称	機　能
F1	英数モードとローマ字かな変換モードを切り替える
F2	ひらがなをカタカナに変換する
F3	半角文字に変換する
F4	入力されたJISコードを文字に変換する
F5	すべての文節を確定する
↲	変換行の先頭の文節を確定する
∧	変換行の先頭の文節の区切りを伸ばす
∨	変換行の先頭の文節の区切りを縮める
＜ ＞	変換カーソルを左右に移動する　（変換中は未変換に戻す）
SPACE	文節変換する　（次候補文字を表示する）
ESC	入力した文字を取り消す　（変換中は未変換に戻す）
SHIFT ＋ ＜ ＞	変換行の先頭または最後に変換カーソルを戻す
SHIFT ＋ SPACE	空白（スペース）の入力

※「文節」は、単語にならない範囲で最も短く文章を区切った単位

※「確定」は、変換された文字をパソコンに入力すること。

文章の入力・変換・確定のしかた

連文節変換での変換行への文字の入力は、文章単位で行います。一度に変換行に入力できる文字数は、ひらがなで最大14文字までです。（だく音、半だく音を含む文字の場合、13文字までです。）

●入力

下のように変換行に文章を入力してください。

ぼくはぱそこんがだいすきだ。｜――変換カーソル（前景色）

このように、白地に黒で表示されている文字を**「未変換文字」**と呼びます。

◆キーを押しまちがえたときは、次のようにして訂正します。

・入力した文章をすべて取り消すときは、ESC（**取消**）を押す。

・まちがえた文字だけを訂正するときは、左向きのカーソルキー（＜）を押して変換カーソル（未変換文字の上では色で表示）をまちがえた文字に合わせ、DEL（**削除**）を押してまちがえた文字を消し、正しい文字を入力する。

ぼくはぱそこんかだいすきだ　──変換カーソル

⬇

DEL

⬇

ぼくはぱそこんだいすきだ　──変換カーソル

⬇

G A（または か ゛）

⬇

ぼくはぱそこんがだいすきだ　──変換カーソル

・文字を挿入するときは、挿入する位置の後ろの文字に変換カーソルを合わせ、挿入したい文字を入力する。
・ BS （**後退**）を押すと、右端の文字から順に一文字ずつ消すことができる。

> **メモ：** 訂正がすんだら、▷ を押して変換カーソルを右端に戻してください。カーソル位置で文節を区切るため、そのままでは正しく変換できないことがあります。

◆読みを忘れた文字や、キーボードに無い記号などを入力するときは、ＪＩＳコード入力と呼ばれる方法を使います。ＪＩＳコード入力のしかたは、106ページ から説明しています。

●変換

未変換文字を漢字に変換するには、 SPACE （**変換**）を押します。

僕はパソコンが大好きだ

変換行に表示されていた文字が一度に漢字やカタカナに変換されます。
先頭の文節だけが、青いバックに白い字で表示され、他の部分は白地に黒で表示されます。

●確定

すべての文字が正しく変換されているときは、 F5 （**全確定**）を押し、変換行の文字を入力カーソルの位置に表示します。

◆正しく変換されていないときは、次のようにして変換し直します。

・先頭の文節からまちがっているときは、次のどれかの操作をする。
　文節の区切りかたが正しいとき……正しい文字が表示されるまで［SPACE］（**変換**）を押す。
　文節の区切りかたが短すぎるとき…正しく文節が区切られるまで**上向き**のカーソルキー（［△］）を押す。
　文節の区切りかたが長すぎるとき…正しく文節が区切られるまで**下向き**のカーソルキー（［▽］）を押す。

・先頭の文節は正しく変換されているが、その後の文字がまちがっているときは、［↵］（**部分確定**）を押す。
先頭の文節だけが入力カーソルの位置に表示され、次の文節が青いバックに白で表示される。
まちがっている文字がある文節が青いバックに表示されるまで、［↵］を押し続ける。あとは先頭の文節からまちがっているときと同じ操作をする。

・変換してから入力のまちがいに気付いたときは、［ESC］（**取消**）または［◁］［▷］を押して変換を取り消し、まちがっている文字を訂正する。訂正がすんだら変換カーソルを右端に戻し、再び変換する。

・カタカナに変換したい文字がカタカナにならなかったときは、次のようにする。
変換カーソルをカタカナにする文節の先頭の文字に合わせ、その前までの文節を変換し、［↵］を押して部分確定する。
変換カーソルをカタカナにする文節の直後の文字に合わせ、［F2］（**カタカナ変換**）を押す。

●読みを知らない漢字や記号の入力のしかた（ＪＩＳコード入力）

音読みを忘れた漢字やキーボードに無い記号を入力するときは、**「ＪＩＳコード」**と呼ばれる文字の番号を変換行に入力し、目的の文字に変換します。
次の手順で入力してください。

1 パソコンの説明書の、「ＪＩＳ第１水準漢字コード表」または「ＪＩＳ第２水準漢字コード表」で目的の文字のＪＩＳコードを調べる。（ただし、非漢字については、本書の205ページ参照）

※　ＪＩＳコードは、16進数４桁であらわします。（16進数⇨55ページ）

例）　「漢」（JIS第１水準漢字：かん）…………３４４１

2 [CTRL] を押しながら [SPACE] を押して、変換行を表示させる。

3 [F1] を押して、英数・かなモードに切り替える。

4 JISコードを入力する。（[3][4][4][1]）

変換行に、入力したコードが表示される。

5 [F4] （**JISコード変換**）を押す。

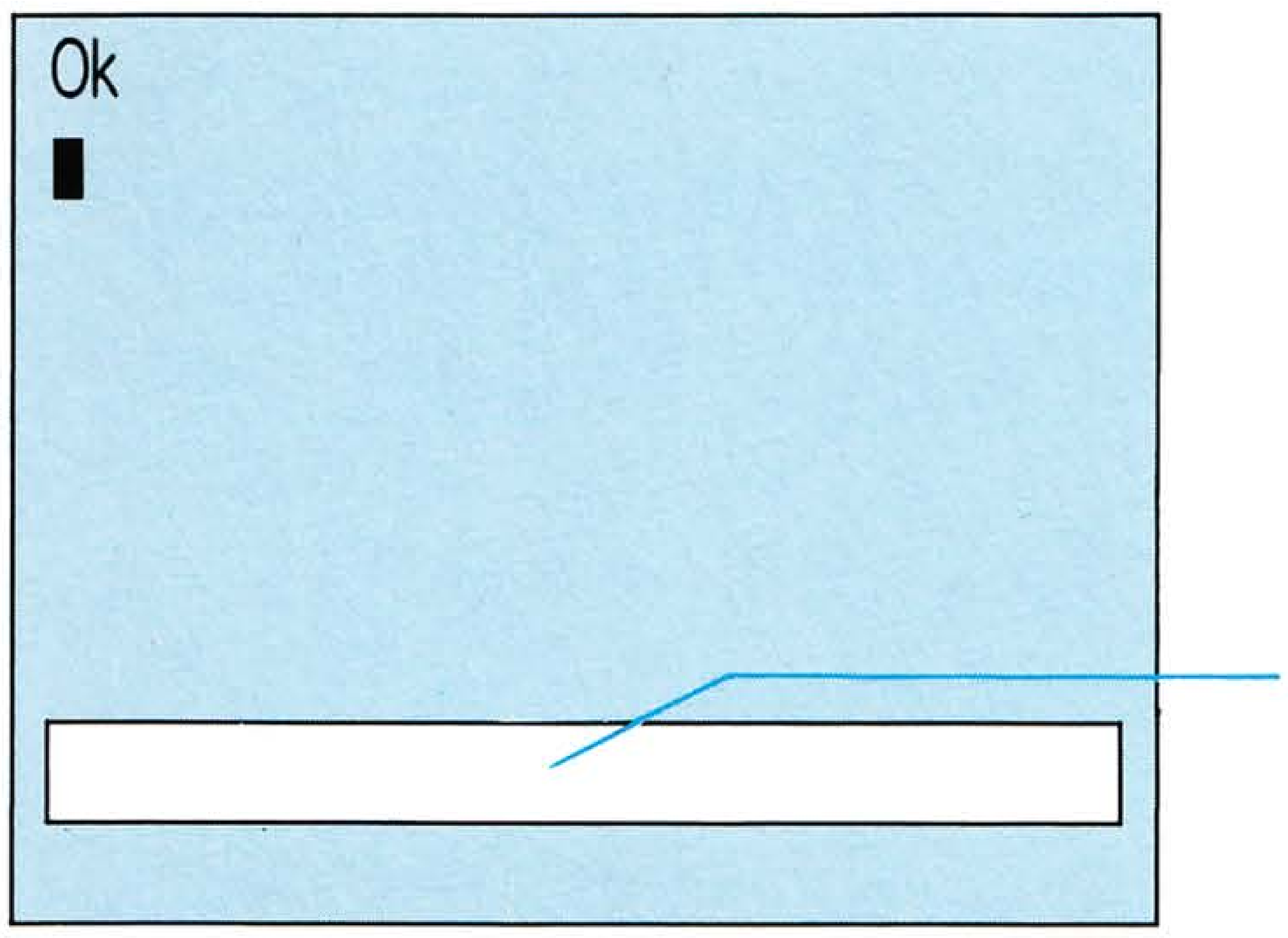
入力したコードにあたる文字が、変換行の左端に表示される。

6 ↵ （**部分確定**）を押して、表示された文字を確定する。

単漢字変換による文字の入力のしかた

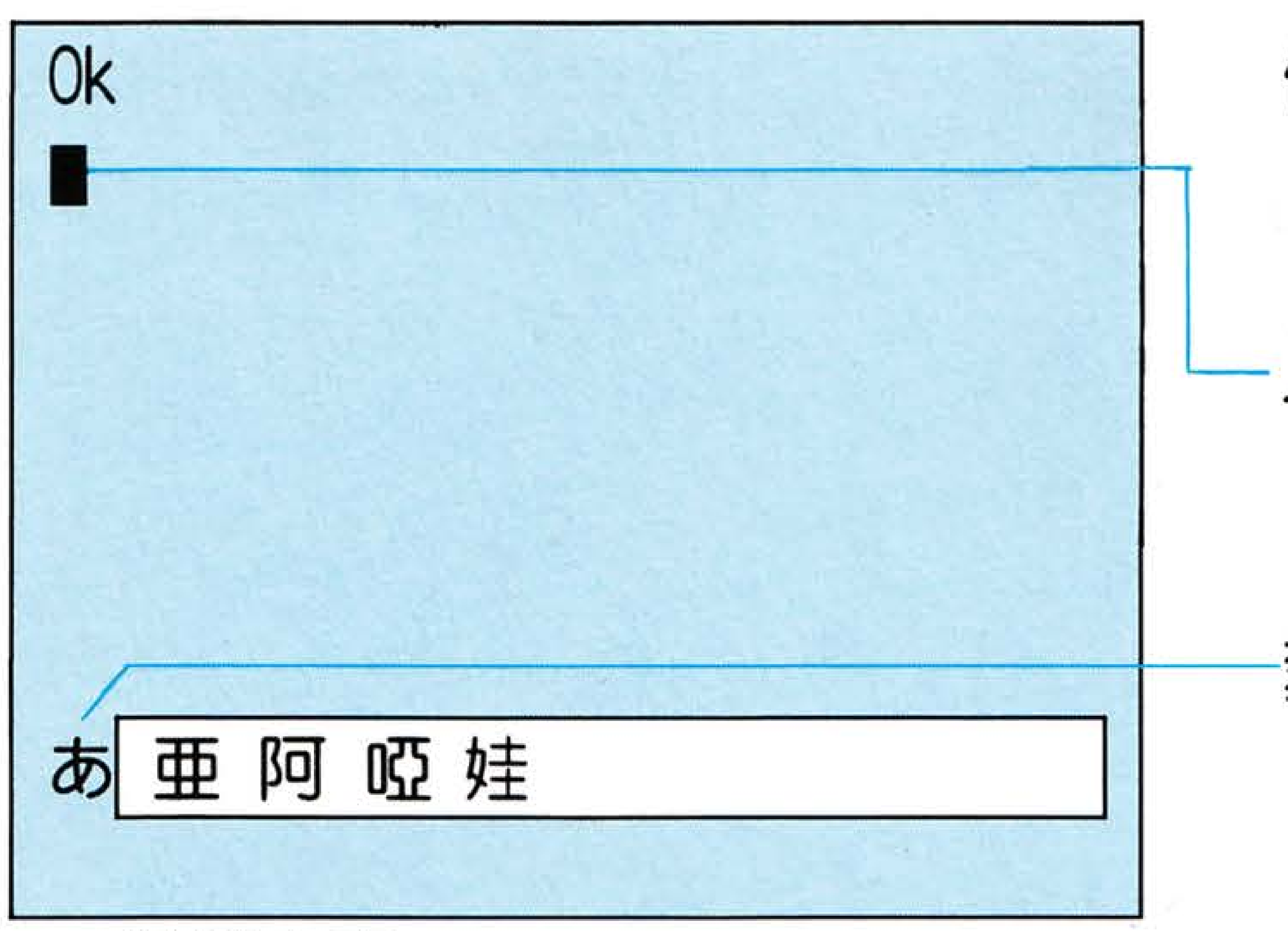
単漢字変換は、連文節変換用ソフトを内蔵していない機種のＭＳＸ２+ パソコンでだけ行います。連文節変換用ソフトを内蔵している機種のＭＳＸ２+ パソコンでは、連文節変換で漢字を入力します。（⇨101ページ）

・ CTRL を押しながら SPACE を押したとき、画面の一番下の行が前景色で窓のように表示されます。これを**「変換ウィンドウ」**と呼びます。



・変換ウィンドウが表示されているときに、ひらがなまたはカタカナを入力すると、入力したひらがなまたはカタカナが変換ウィンドウの左端に背景色のバックに前景色で表示され、その文字を音読みとする漢字が右側に前景色のバックに背景色で一列に表示されます。

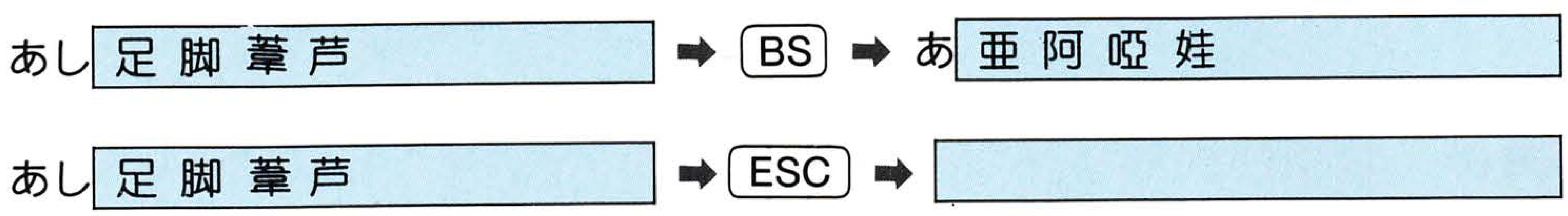
例）ひらがなの「あ」のキーを押したとき



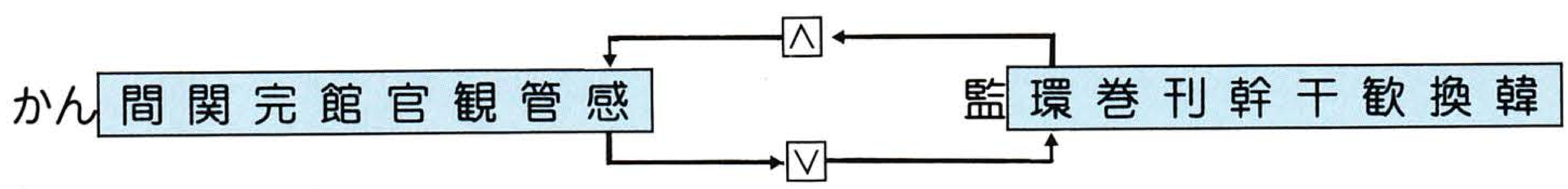
漢字の入力のしかた

変換ウィンドウに表示された漢字を**「候補文字」**と呼びます。
このときさらに文字キーを押すと、押したキーの文字が左端のかな文字の右側に続けて表示され、右側の候補文字が両方のかなを音読みとする漢字に入れ替わります。あてはまる文字がないときには、何も表示されません。

◆キーを押しまちがえたときは、 BS を押すと最後から一文字ずつ消すことができます。
ESC を押すと、すべての文字を取り消すことができます。
ただし、どれかのカーソルキーを押したあとでは、この訂正はできません。

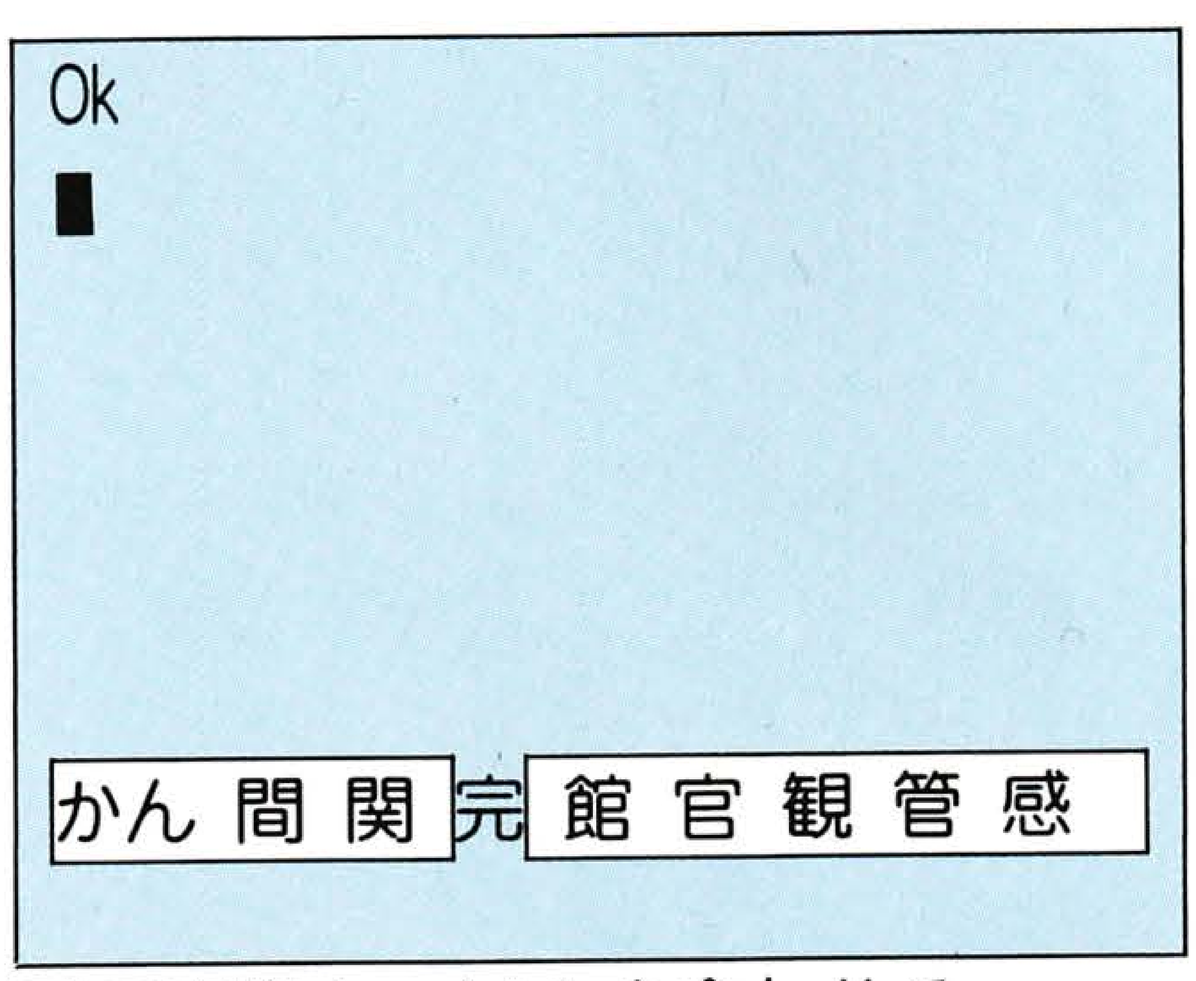


◆表示されている候補文字の中に入力したい漢字がないときは、下向きのカーソルキー（▽）を押してみます。同じ読みの漢字がまだあれば、候補文字が次の一組みと入れ替わります。
上向きのカーソルキー（ △ ）を押すと、その前に表示した一組みに戻ります。

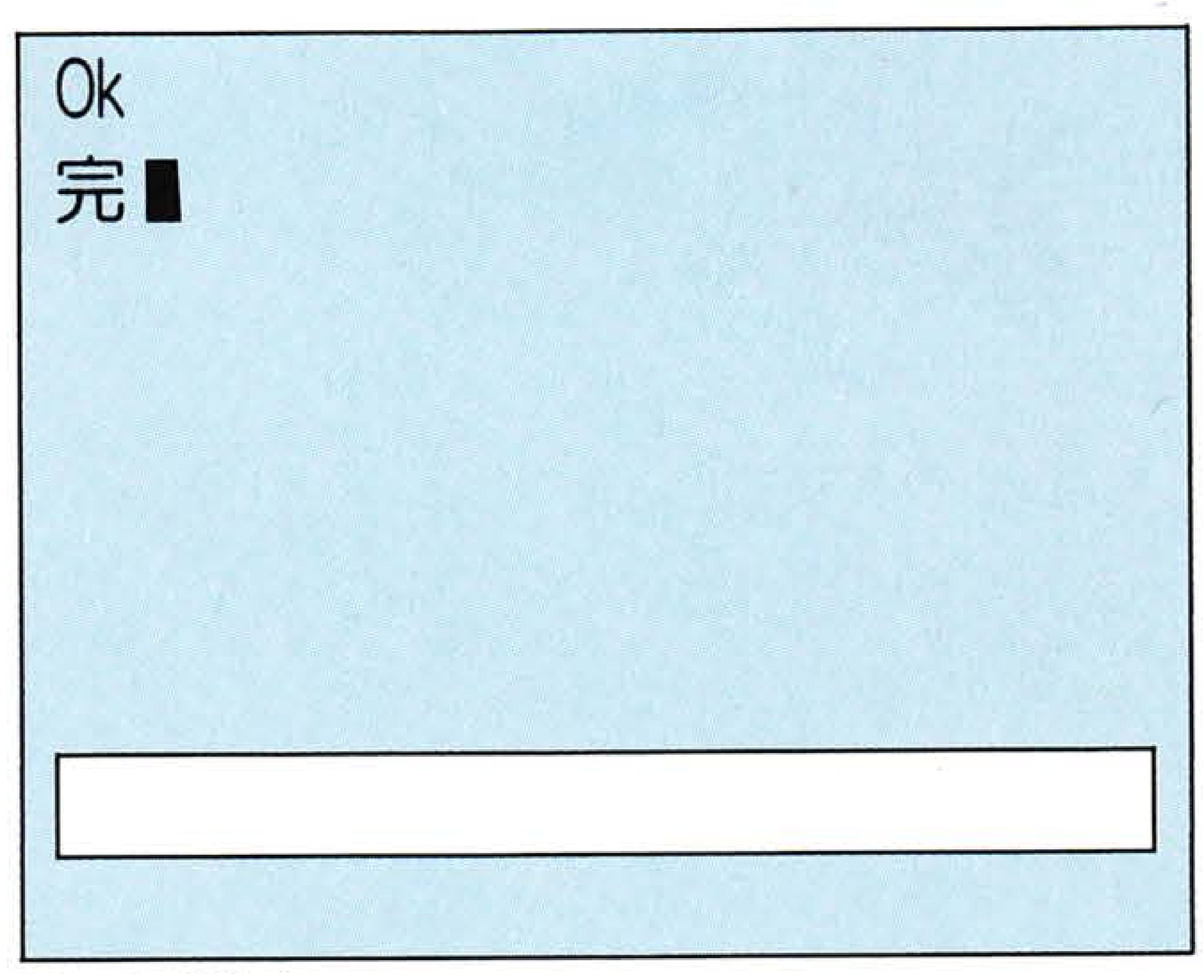


▷を押すと、変換ウィンドウの背景色の部分が右に一文字ずつ移って行き、ウィンドウの右端まで来ると、次の一組みが表示されます。

入力したい漢字が表示されたら、左右のカーソルキーを使って、変換ウィンドウの中の背景色の部分（変換カーソル）をその漢字に合わせ、 ↵ を押します。



▷で変換カーソルを合わせる



↵ を押す

・漢字と同じ大きさ（全角）の、英文字、数字、ひらがな、カタカナ、記号を入力したいときは変換ウィンドウが表示されているときにそれらの文字のキーを押し、そのまま ⏎ を押します。（全角のカタカナを入力するときは、ＣＡＰＳとかなのランプが点灯していることが必要です。）

> **メモ：**ＢＡＳＩＣの命令や行番号、数値、演算子、変数名、ファイル名などに全角文字を使うことはできません。半角文字で入力してください。

・グラフィック文字（●や｜など）は、次に述べる**「JIS コード入力」**という入力方法で入力してください。（一部の文字は、半角文字と形が異なります。）

●読みを知らない漢字や記号の入力のしかた（ＪＩＳコード入力）

音読みを忘れた漢字やキーボードに無い記号を入力するときは、「ＪＩＳコード」と呼ばれる文字の番号を変換ウィンドウに入力し、目的の文字に変換します。
次の手順で入力してください。

1　パソコンの説明書の、「ＪＩＳ第１水準漢字コード表」または「ＪＩＳ第２水準漢字コード表」で目的の文字のＪＩＳコードを調べる。（ただし、非漢字については、本書の205ページ参照）
※　ＪＩＳコードは、16進数４桁であらわします。（16進数⇨55ページ）
例）　「漢」（JIS第１水準漢字：かん）‥‥‥‥３４４１

2　CTRL と SPACE を押して、変換ウィンドウを画面に表示する。

3　CTRL を押しながらＪＩＳコードを入力する。（ 3 4 4 1 ）
※通常は、入力した数字は表示されません。（⇨次ページ）
変換ウィンドウに、入力したコードの文字とそれに続くコードの文字が表示される。

漢 潤 潅 環 甘 監 看 竿 管

※入力したコードに続くコードの文字を最大128個まで、上下のカーソルキーを使って表示させることができます。

4　⏎ を押して、入力カーソルの位置に入力したコードの文字を表示する。

◆JIS第２水準漢字は、JIS第２水準漢字ＲＯＭをもたないパソコンでは画面に表示できません。そのようなパソコンで JIS第２水準漢字のコードを入力した場合、下のように変換ウィンドウにJISコードが表示されます。

53	53	53	53	53	53	53	53	54
77	78	79	7A	7B	7C	7D	7E	21

※JISコード5377を入力した場合

・この状態で作成したプログラムなどをディスクに保存し、他の JIS第２水準漢字ＲＯＭを内蔵するＭＳＸ２+ パソコンで呼び出すと、その JISコードにあたる文字が画面に表示されます。また、そのようなプログラムなどを JIS第２水準漢字ＲＯＭを内蔵するプリンタで印刷するとその JISコードにあたる文字が印刷されます。

単漢字変換での各キーの機能

キーの名称	機　能
CTRL ＋ SPACE	変換ウィンドウの表示・消去を切り替える
BS	変換ウィンドウに入力した文字を最後から一文字ずつ消す
ESC	変換ウィンドウの全文字を消す（カーソルキーの操作後は無効）
◁ ▷	変換カーソルを左右に移動する
▽ △	候補文字を入れ替える
↵	変換カーソルの位置の候補文字を、入力カーソルの位置に表示
CTRL	JISコード入力のとき、押し続ける

漢字モードの種類

ＢＡＳＩＣの漢字モードには、画面に表示する文字のサイズと表示のしかたによって４つの種類があります。
これらは、ＣＡＬＬ　ＫＡＮＪＩ命令を実行させるときに選びます。

・**ＣＡＬＬ　ＫＡＮＪＩ０**……文字サイズはヨコ16×タテ16ドット。
（または CALL KANJI）

・**ＣＡＬＬ　ＫＡＮＪＩ１**……文字サイズはヨコ12×タテ16（または12）ドット。

・**ＣＡＬＬ　ＫＡＮＪＩ２**……文字サイズはヨコ16×タテ16（または12）ドット。
インターレースモードで表示される。

・**ＣＡＬＬ　ＫＡＮＪＩ３**……文字サイズはヨコ12×タテ16ドット。
インターレースモードで表示される。

※「ＫＡＮＪＩ」と数字のあいだは、間隔をあけないでください。間隔をあけて入力するとエラーになります。
※インターレースモードとは、１つの画面に表示できる文字の行の数を２倍にする表示方法のことですが、一般の家庭用テレビではちらつきが大きく、実用的ではありません。

以後の説明では、ＣＡＬＬ　ＫＡＮＪＩ０およびＣＡＬＬ　ＫＡＮＪＩ１だけを使用します。

漢字モードの画面

ＢＡＳＩＣの英数・かなモードにテキスト画面とグラフィック画面があるように、漢字モードにも「漢字テキスト画面」と「漢字グラフィック画面」があります。
漢字モードでの画面の種類は、下の表のようになっています。
（この表は、漢字モード時の各画面モードの仕様をしめしています）

<table>
<tr><th>モード</th><th>スクリーン No.</th><th>文字表示</th><th>図形表示</th><th>表示色</th><th>色指定</th></tr>
<tr><td rowspan="2">漢字テキスト</td><td>0</td><td>半角文字最大８０桁＊１
全角文字最大４０桁</td><td>表示できない</td><td rowspan="2">５１２色中の１６パレット中の２色</td><td rowspan="2">前景色
背景色
周辺色</td></tr>
<tr><td>1</td><td>半角文字最大８０桁＊１
全角文字最大４０桁</td><td>スプライトのみ可能
（スプライト座標使用）＊２</td></tr>
<tr><td rowspan="7">漢字グラフィック</td><td>2</td><td>半角文字最大４０桁＊３
全角文字最大２０桁</td><td>横２５６×縦１９２ドット</td><td rowspan="4">５１２色中の１６パレット</td><td>水平８ドットを１単位とする領域ごとに２色</td></tr>
<tr><td>3</td><td>半角文字最大１０桁＊３
全角文字最大　５桁</td><td>横２５６×縦１９２ドット
（６４×４８ブロック）＊４</td><td>１ブロックごとに色指定可能</td></tr>
<tr><td>4</td><td rowspan="2">半角文字最大４０桁＊３
全角文字最大２０桁</td><td>横２５６×縦１９２ドット</td><td>水平８ドットを１単位とする領域ごとに２色</td></tr>
<tr><td>5</td><td>横２５６×縦２１２ドット</td><td rowspan="4">１ドットごとに色指定可能</td></tr>
<tr><td>6</td><td>文字表示には向かない</td><td rowspan="2">横５１２×縦２１２ドット</td><td>５１２色中の４パレット</td></tr>
<tr><td>7</td><td>半角文字最大８０桁＊３
全角文字最大４０桁</td><td>５１２色中の１６パレット</td></tr>
<tr><td>8</td><td rowspan="2">半角文字最大４０桁＊３
全角文字最大２０桁</td><td rowspan="4">横２５６×縦２１２ドット</td><td>２５６色固定</td></tr>
<tr><td rowspan="3">自然画表示</td><td>10</td><td>自然画表示色、１２，４９９色</td><td rowspan="3">＊５）</td></tr>
<tr><td>11</td><td rowspan="2">文字表示には向かない。</td><td>自然画表示色、１９，２６８色</td></tr>
<tr><td>12</td><td>自然画表示色、１９，２６８色</td></tr>
</table>

＊１）ＣＡＬＬ　ＫＡＮＪＩ１、ＷＩＤＴＨ　８０実行時の数値。
＊２）スプライト座標は、Ｘは－３２～２５５、Ｙは－３２～１９１（スクリーン５以上では、－３２～２１１）。
＊３）ＣＡＬＬ　ＫＡＮＪＩ１実行時の数値。プログラムモードでのみ表示可能。
＊４）１ブロックは４×４ドットの領域。
＊５）スクリーン１１・１２では、図形表示に２５６色分の色コードを指定できますが、色表示の方法が他の画面と異なるため、同じコードによる色指定をしても他の画面とは異なる色になったり、まったく表示されなかったりします。図形表示はスクリーン１０までの画面モードでしてください。

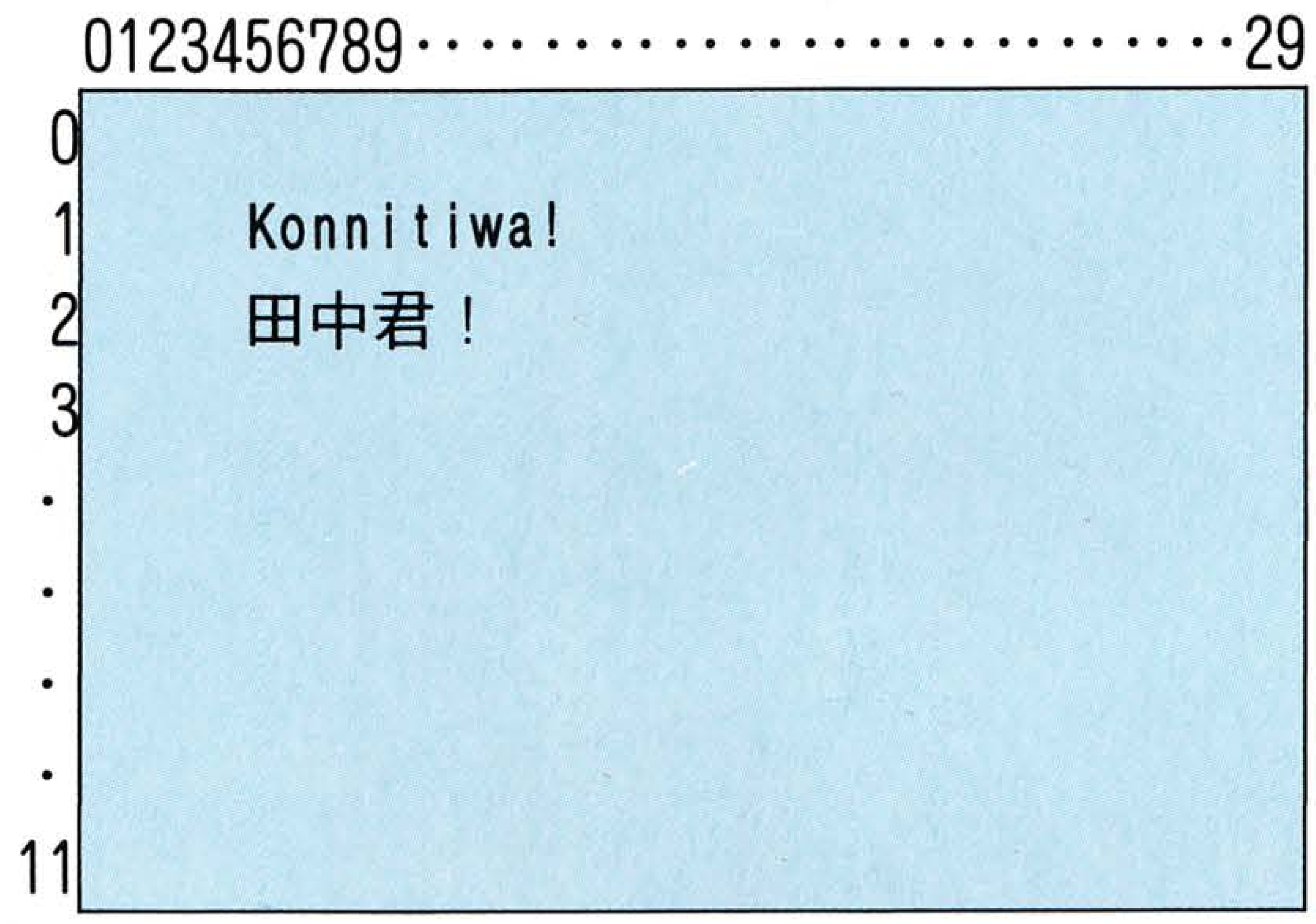
漢字モードの文字表示位置座標

漢字モードでは、漢字テキスト画面、漢字グラフィック画面のどちらでも、ＬＯＣＡＴＥ文やＰＲＩＮＴ文で画面の指定した位置に文字を表示することができます。

このとき、画面の表示位置は、半角文字一文字を単位とする桁と行で指定します。

（**例**：ＫＡＮＪＩ０でヨコ座標の指定を５にしたときの表示位置）

```
10LOCATE 5、2↵
20PRINT"田中君！"↵
```



漢字モードでは、漢字テキスト画面、漢字グラフィック画面のどちらでも、文字の表示位置を上のように半角文字単位で指定します。

漢字で住所氏名を表示しよう

漢字テキストモードを使ったプログラムです。
ＩＮＰＵＴ文で住所氏名を漢字で入力し、色を変えて画面に表示します。



※プログラムを止めるときは、［CTRL］を押しながら［STOP］を押します。

※プログラムの漢字以外の部分は、半角文字で入力してください。全角文字で入力するとエラーになります。

リストと解説

プログラム	解説
`1 'save"ｼﾞｭｳｼｮ.bas"`	
`100 CALL KANJI`	漢字モード０に切り替える
`110 COLOR 15,4`	表示色の初期設定
`120 CALL CLS`	漢字モードでの画面消去
`130 PRINT:COLOR 15`	前景色の再設定（白）
`140 INPUT "氏名";NM$`	氏名住所の入力
`150 INPUT "住所";J$`	（同上）
`160 PRINT:COLOR 7`	改行：前景色の変更（淡い青）
`170 PRINT NM$`	氏名住所の表示
`180 PRINT J$`	（同上）
`190 GOTO 130`	

参　考

・漢字プリンタを接続している場合、160行〜180行のＰＲＩＮＴ文をＬＰＲＩＮＴ文に変えることにより、入力した氏名・住所を印刷することができます。
・131ページ にかんたんな住所録作成のサンプルプログラムを掲載しています。参考にしてください。

新しい命令の解説

漢字モードの文字列

漢字モードでは、文字列に全角文字を使うことができます。
（100,140,150,170,180行）

英数・かなモードでは、１つの文字列に 255文字までの半角文字を使うことができました。
漢字モードでは、１つの文字列に全角文字を 127文字まで使うことができます。
（全角文字１文字は、半角文字２文字にあたります）

ただし、次の場合には文字列に全角文字を使うことができませんから注意してください。

例）

```
OPEN "GRP:" AS #1:PRINT#1,"文字列"
```

CALL CLS（コール・シーエルエス）

漢字モードの漢字テキスト画面では、英数・かなモードで使えたＣＬＳ命令が使えません。

（使うとエラーになります）
漢字テキスト画面を消去するときは、ＣＡＬＬ　ＣＬＳ命令を使います。
ＣＡＬＬ　ＣＬＳは、漢字グラフィック画面でも、英数・かなモードでも使えます。（120行）
※漢字グラフィック画面では、ＣＬＳ命令も使えます。

漢字モードの文字表示色

英数・かなモードでは、前景色の指定を変えると画面に表示されているすべての文字の色が変わります。同時に画面に表示する文字の色が１色だけに限られているからです。

漢字モードでは、１文字単位で表示色を指定できます。

前景色の指定を変えても、先に表示した文字の色は変わりません。（130行、160行）

漢字グラフィック画面で文字表示

漢字グラフィック画面で、図形に漢字を重ねて表示するプログラムです。



※プログラムを止めるときは、CTRL を押しながら STOP を押します。

リストと解説

```
1 'save"ｶﾝｼﾞGRP.bas"
100 CALL KANJI:SCREEN 5
110 CALL PALETTE:COLOR ,4
120 FOR R=5 TO 100 STEP 5
130   CIRCLE (125,103),R,8
140 NEXT R
150 LOCATE 9,6:PRINT "漢字B
ASIC"
160 GOTO 160
```

- 100: スクリーン5の画面に切り替える
- 120〜140: 5から 100まで5ずつ半径を増しながら、赤で同心円を描く
- 150: 同心円に重ねて文字を表示する

新しい命令の解説

漢字グラフィック画面の文字表示

漢字グラフィック画面に文字を表示するときには、漢字テキスト画面と同じく、次の２つの命令を使用します。

LOCATE………表示位置の指定
PRINT…………文字列の表示

・表示位置の指定に使用する位置座標も、漢字テキスト画面と同じく、半角文字単位の桁と行で指定します。
・漢字グラフィック画面では、WIDTH文は使用できません。
１行に表示できる文字数（桁数）は、最大で１行20文字を限度として、漢字モードの種類による文字のドット数と画面モードによる画面のヨコ方向のドット数で決定されます。

例）

漢字モード０（ヨコ16ドット）で画面モードがスクリーン５（ヨコ 256ドット）のとき
256÷16＝16　で　１行の表示文字数は、全角文字16桁。（半角文字32桁）
漢字モード１（ヨコ12ドット）で画面モードがスクリーン７（ヨコ 512ドット）のとき
512÷12＝42 だが１行の表示文字数は、全角文字40桁。（半角文字80桁）

おまけのプログラム

```
1 'save"ﾃﾝﾁ.bas"
100 CALL KANJI:SCREEN 5
110 COLOR 4,0,0:CALL CLS
120 LOCATE 12,5
130 PRINT "天地無用"
140 FOR N=0 TO 79
150   X=96+N
160   FOR M=0 TO 15
170     COPY (X,80+M)-(X,80+M
) TO (X,112-M)
180   NEXT M
190 NEXT N
200 GOTO 200
```

漢字グラフィック画面を使ったおまけのプログラムです。
特に解説はしませんが、実行には少し時間がかかります。

※プログラムを止めるときは、CTRL を押しながら STOP を押してください。

※COPY（コピー）命令のこのような使いかたについては、この本では説明していません。興味がある人は、市販のMSX　BASICの解説書で調べてみてください。

「入門」のおわりに

さて、ＢＡＳＩＣについての説明は、これで一応おしまいです。お疲れさまでした。
どうですか？　思ったよりカンタンだったでしょう？

この本では、ＢＡＳＩＣのもつすべての命令について説明することよりも、むしろＢＡＳＩＣに親しんでもらうことに重点をおきました。
これまでに説明してきたことは、ＢＡＳＩＣがもつたくさんの命令や関数のほんの一部分に過ぎません。
言い換えれば、あなたはいまＢＡＳＩＣへの「入門」を終わったばかりなのです。

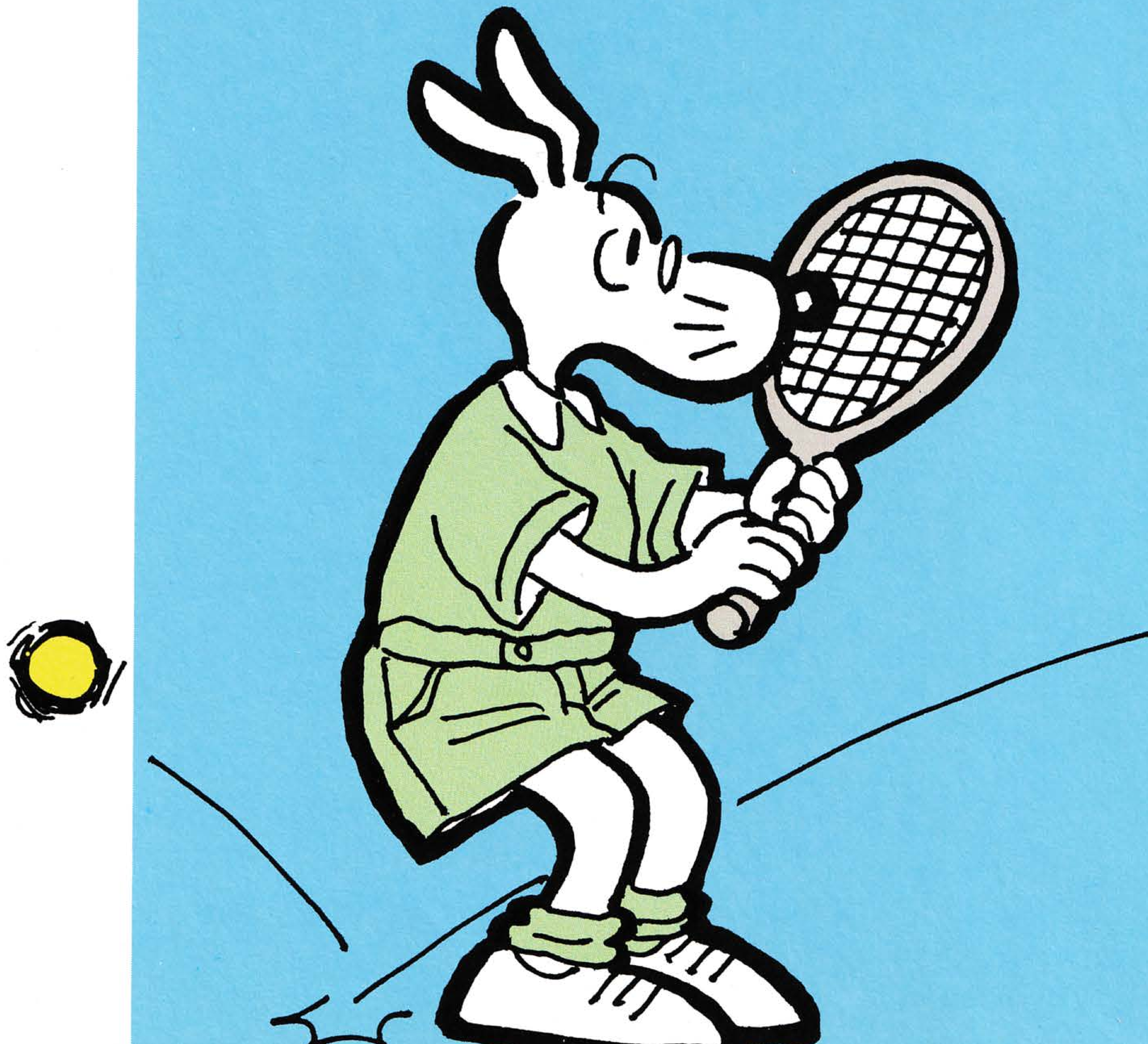
「もっとＢＡＳＩＣの勉強をしたい」という人は、この本の 171ページから始まる「ＢＡＳＩＣリファレンス表」や、書店やパソコンの専門店で販売されているＭＳＸ　ＢＡＳＩＣの解説書などを使って、どうぞ勉強を続けてください。
また、「この辺で一息入れたいな」という人は、次のページから始まる「第６章　サンプルプログラム集」の中から、好きなプログラムを選んで試してみてください。

ＢＡＳＩＣをマスターするひとつのコツは、あせらず、あきらめずに根気よく続けるということです。「私はＢＡＳＩＣをマスターした」と言える日をめざして、どうかあきらめずに頑張ってください。

第6章

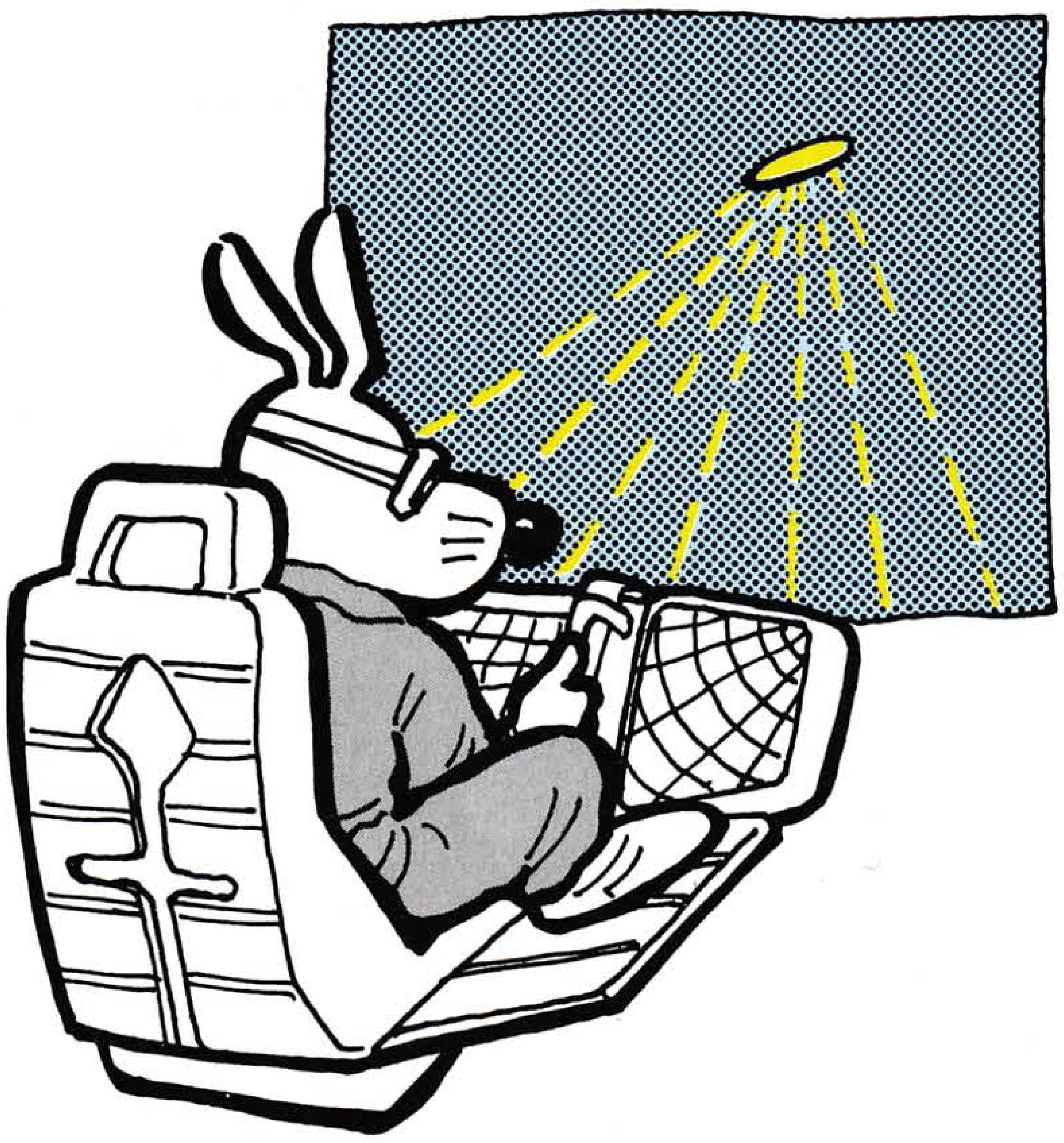
楽しむためのＢＡＳＩＣだという人のためのサンプル集です。

サンプルプログラム集



UFO

上空に飛んでくるＵＦＯを、砲台を左右に動かしながらミサイルで撃ち落とすゲームです。うまくＵＦＯを撃ち落とすと得点になりますが、はずしてしまうと得点がマイナスされます。１機のＵＦＯに対して１発しかミサイルを撃てませんから、慎重に狙ってください。



砲台の移動は左右のカーソルキーを、ミサイルの発射は SPACE を押します。

プログラムを途中で止めるときは、CTRL を押しながら STOP を押したあと、ダイレクトモードで次の命令を実行させてください。

SCREEN 5
（スプライトパターンの消去）

```
1 'save"UFO.BAS"
100 CALL ANK
110 SCREEN 1,1:WIDTH 29:COLOR
:COLOR 15,4,7:CLS:KEY OFF
120 FOR K=0 TO 2
130   B$=""
140   FOR J=0 TO 7
150     READ A$
160     B$=B$+CHR$(VAL("&B"+A
$))
170   NEXT J
180   SPRITE$(K)=B$
190 NEXT K
200 LOCATE 3,15:PRINT"とくてん"
210 ON SPRITE GOSUB 470
220 SPRITE ON
230 YN=-18:YY=209
240 X=150:T=1
250 DX=10
```

- 100：ＢＡＳＩＣを英数・かなモードに
- 110：画面を初期設定する
- 120〜190：スプライトパターンを定義する
- 210：スプライト同志が衝突したら 470行へ
- 220：以後、スプライトの衝突をチェックする
- 230〜250：各種の変数の値を初期設定する

```
260 PUT SPRITE 0,(XX,YY),W,0
270 IF W=4 THEN W=10 ELSE W=4

280 XX=XX+DX:IF XX<239 THEN 3
10
290 XX=0:DX=RND(1)*8+5:YY=RND
(1)*100:T=INT(T-1):LOCATE 6,1
6:PRINT T:FG=1
300 N=N+1:IF N>50 THEN 410
310 ST=STICK(0):IF ST=0 THEN
GOTO 360
320 IF ST=3 THEN X=X+16
330 IF X>239 THEN X=239
340 IF ST=7 THEN X=X-16
350 IF X<0 THEN X=0
360 PUT SPRITE 1,(X,174),8,1
370 PUT SPRITE 2,(X2,Y2),10,2

380 Y2=Y2-16:IF Y2>-18 THEN 2
60
390 Y2=-18:IF FG*STRIG(0)=0 T
HEN 260
400 FG=0:Y2=157:X2=X+2:GOTO 2
60
410 '* ｹﾞｰﾑ ｵｰﾊﾞｰ *
420 SCREEN 2:SCREEN 1
430 LOCATE 3,7:PRINT"ｹﾞｰﾑ ｵｰﾊ
ﾞｰ !"
440 LOCATE 5,10:PRINT"とくてん=";
T
450 PLAY"O4V8L4DCFG8O5C8O4AG8
O5C8O4AFGDC"
460 KEY ON:PRINT:END
```

- 260〜290：ＵＦＯの移動と色の変化の処理
- 300：ゲームが終了したかどうかの判定
- 310〜360：カーソルキーと SPACE が押されているかどうかを調べ、カーソルキーが押されていたら砲台を左右に動かし、 SPACE が押されていたら280行へ
- 370〜380：ミサイルの発射と移動の処理
- 410〜460：ゲーム終了時の処理

```
470 '* ﾒｲﾁｭｳ *
480 IF YY=209 THEN RETURN
490 T=T+DX-3:PLAY "O3S9M2000C
"
500 YY=209:Y2=-18
510 RETURN
520 '* UFO ﾊﾟﾀｰﾝ *
530 DATA 00011000
540 DATA 00111100
550 DATA 01111110
560 DATA 11111111
570 DATA 00100100
580 DATA 00100100
590 DATA 01000010
600 DATA 10000001
610 '* ﾎｰﾀﾞｲ ﾊﾟﾀｰﾝ *
620 DATA 00111100
630 DATA 00111100
640 DATA 00111100
650 DATA 00111100
660 DATA 00111100
670 DATA 00111100
680 DATA 11111111
690 DATA 11111111
700 '* ﾐｻｲﾙ ﾊﾟﾀｰﾝ *
710 DATA 00000000
720 DATA 00000000
730 DATA 00010000
740 DATA 00010000
750 DATA 00010000
760 DATA 00010000
770 DATA 00000000
780 DATA 00000000
```

ミサイルがＵＦＯに当たったときの処理（470〜510）

ＵＦＯと砲台、ミサイルのスプライトパターン（520〜780）

ヘリコプター

あなたはヘリコプターの操縦を習っている訓練生です。いじわるな教官から、燃料切れ寸前のヘリコプターで移動するプラットフォームの上に着陸するよう命令されました。みごとに着陸してみせて、教官に「マイッタ！」と言わせましょう。



※ゲームが始まったとき、ヘリコプターはエンストをおこしています。[∧]ですぐに上昇してください。

※なにしろオンボロのヘリコプターですから、そっと着陸してください。いきおいがついたままプラットフォームに着陸すると、バラバラになってしまいます。

※着陸に成功すると、残っている燃料の分だけ得点になります。

```
1 'save"ﾍﾘｺﾌﾟﾀｰ.BAS"
1000 CALL ANK
1010 SCREEN 5,1:COLOR:COLOR 1
5,4,7:CLS
1020 FOR T=0 TO 2
1030   B$=""
1040   FOR N=0 TO 7
1050     READ A$
1060     B$=B$+CHR$(VAL("&b"+
A$))
1070   NEXT N
1080   SPRITE$(T)=B$
1090 NEXT T
1100 OPEN "grp:" AS #1
1110 ON SPRITE GOSUB 1430
1120 TK=0
```

- 1000：ＢＡＳＩＣを英数・かなモードに
- 1010：スクリーン５の画面でスプライトの種類を８×８ドット拡大にし、画面を初期設定する
- 1020〜1090：スプライトパターンを３個定義する
- 1100：グラフィック画面を文字表示可能に
- 1110：ヘリコプターがプラットフォームに衝突したら1430行へ
- 1120：得点をあらわす変数ＴＫの値を０に

```
1130 '--- ｹﾞｰﾑ ｽﾀｰﾄ
1140 X=0:Y=0:U=8:F=0:P=100
1150 X1=-16:Y1=196
1160 PRESET ( 84,8):PRINT #1,
"ﾄｸﾃﾝ" ;TK
1170 PRESET (176,8):PRINT #1,
"ﾈﾝﾘｮｳ";P
1180 SPRITE ON
1190 '--- ﾍﾘｺﾌﾟﾀｰｦ ｳｺﾞ ｶｽ
1200 Y=Y-U:X=X+F
1210 IF X<0   THEN X=  0:F=0
1220 IF X>240 THEN X=240:F=0
1230 IF Y<0   THEN Y=  0:U=0
1240 IF Y>196 THEN Y=196
1250 PUT SPRITE 1,(X,Y),10,1
1260 IF Y=196 THEN GOSUB 1430

1270 PLAY"T255V15O4S8M1000B64
"
1280 '--- ﾌﾟﾗｯﾄﾌｫｰﾑｦ ｳｺﾞ ｶｽ
1290 X1=X1+2:IF X1=256 THEN X
1=-14
1300 PUT SPRITE 2,(X1,Y1),13,
2
1310 '--- ｶｰｿﾙｷｰｦ ｼﾗﾍﾞﾙ
1320 K=STICK(0)
1330 IF K=0 THEN U=U-1:GOTO 1
410
1340 P=P-1:IF P<=0 THEN P=0:U
=U-1:F=0
1350 PRESET (216,8):PRINT #1,
P
1360 IF P=0 THEN 1420
1370 IF 7<K OR  K<3 THEN U=U+
1
```

各変数に設定する初期値

得点と残りの燃料を画面に表示する

以後、スプライトの衝突による割り込み処理を行う

表示する位置を決める

ヘリコプターを表示する

地面にぶつかっていたら、1430行へ

飛行中の音を鳴らす

表示する位置を決める

プラットフォームを表示する

押されているカーソルキーを調べ、各方向への速度と燃料の残量を変える

燃料の残量を表示する

燃料が残っていないなら、1420行へ

```
1380 IF 1<K AND K<5 THEN F=F+
1
1390 IF 3<K AND K<7 THEN U=U-
1
1400 IF 5<K         THEN F=F-
1
1410 '
1420 GOTO 1190                ――1190行に戻る
1430 '--- しょうとつ
1440 SPRITE OFF               ――スプライトの衝突による割り込みを禁止する
1450 IF X<X1-4 OR X1+4<X THEN
 1500                         ――プラットフォームの中心から 4 ドット以上外れ
                                ていたら1500行へ
1460 IF 184<Y             THEN
 1500                         ――勢いがつきすぎていたら1500行へ
1470 '--- ちゃくりく せいこう
1480 PUT SPRITE 1,(X,182),10,
1                             ――プラットフォーム上にヘリコプターを表示する
1490 TK=TK+P:Z$="ﾁｬｸﾘｸ ｾｲｺｳ!"
:GOTO 1550                    ――成功したときの得点と表示を設定する
1500 '--- ちゃくりく しっぱ゛い
1510 PLAY"T120V15O3S9M15000C1
"                             ――爆発音を鳴らす
1520 TK=TK-100+P              ┐
1530 Z$="ﾁｬｸﾘｸ ｼｯﾊﾟｲ!"         ┘―失敗したときの得点と表示を設定する
1540 PUT SPRITE 0,(X,Y+4),9,0 ――爆発のスプライトを表示する

1550 '--- 1ﾗｳﾝﾄﾞ しゅうりょう
1560 PRESET(75,84):PRINT #1,Z
$                             ――結果を表示する
1570 IF 500<TK THEN Z$="  ﾏｲﾘ
ﾏｼﾀ!!  ":GOTO 1650            ――得点がプラス500を越えたときは、別の表示を
                                設定し1650行へ
1580 IF TK<-200 THEN Z$="ｷﾐ ﾊ
ﾗｸﾀﾞｲｼﾀ!":GOTO 1650           ――得点がマイナス200を越えたときは、別の表示
                                を設定し1650行へ
1590 FOR T=0 TO 999:NEXT      ――表示を読むための時間をあける
```

第6章 サンプルプログラム集

```
1600 PUT SPRITE 0,(0,217),,0
1610 PUT SPRITE 1,(0,217),,1
1620 PUT SPRITE 2,(0,217),,2
1630 CLS
1640 RETURN 1130
1650 '--- ｹﾞｰﾑ しゅうりょう
1660 PRESET(80,102):PRINT#1,Z
$
1670 PRESET ( 84,8):PRINT #1,
"ﾄｸﾃﾝ" ;TK
1680 FOR I=0 TO 4999:NEXT
1690 END
1700 '--- ｽﾌﾟﾗｲﾄ ﾃﾞｰﾀ
1710 DATA 00010000
1720 DATA 01011010
1730 DATA 01111110
1740 DATA 11111111
1750 DATA 00111111
1760 DATA 01011010
1770 DATA 11001001
1780 DATA 00001000
1790 '
1800 DATA 01111111
1810 DATA 00001100
1820 DATA 11010010
1830 DATA 11011111
1840 DATA 01111111
1850 DATA 00111110
1860 DATA 00100100
1870 DATA 01111110
1880 '
1890 DATA 00000000
1900 DATA 11111111
1910 DATA 11000011
1920 DATA 10100101
1930 DATA 10011001
1940 DATA 10100101
1950 DATA 11000011
1960 DATA 11111111
```

1600–1640: すべてのスプライトを消し、1130行へ戻ってゲームを再開する

1660–1690: 最終結果と最終得点を表示し、少し時間をあけてからプログラムの実行を終了する

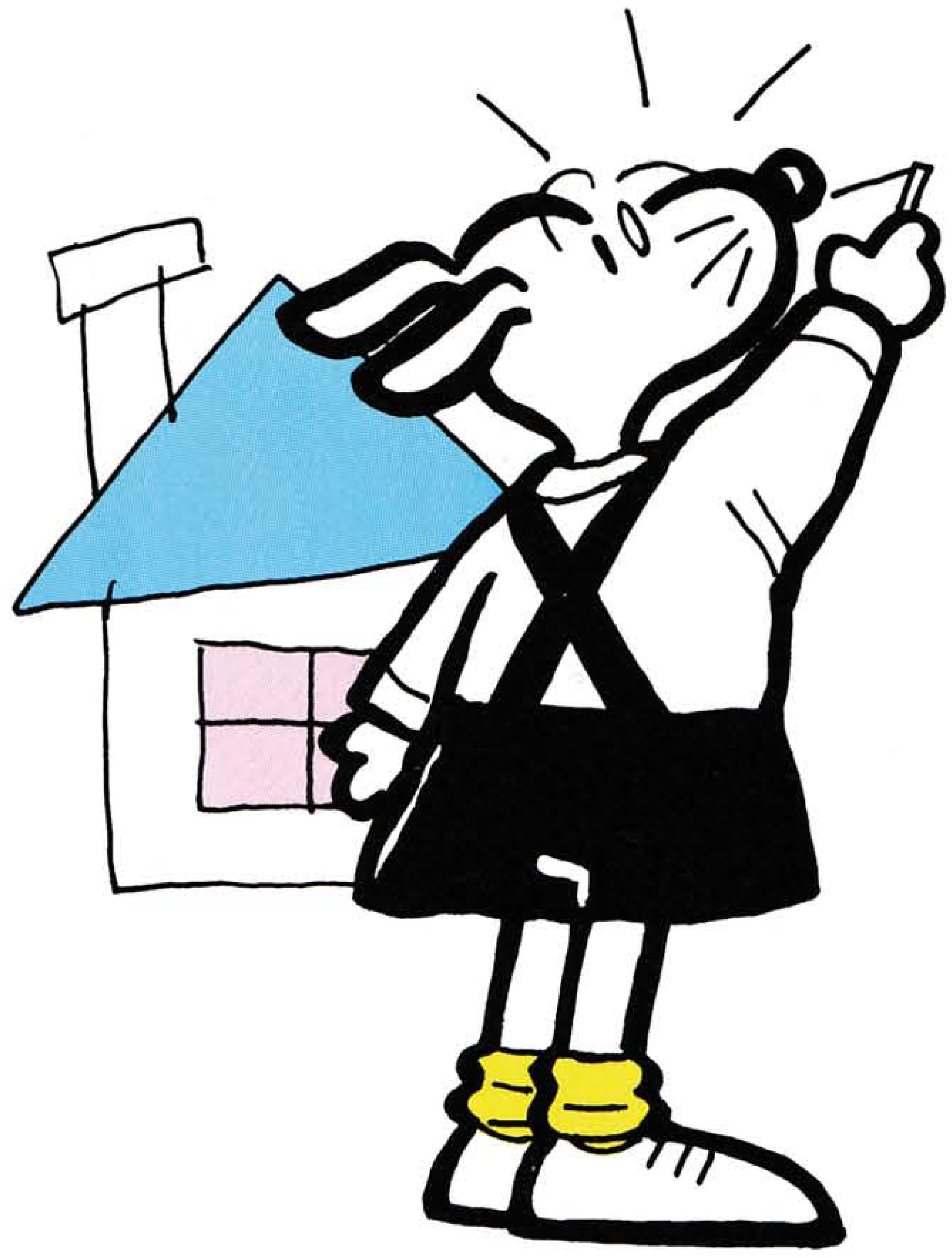
1710: 爆発のスプライトパターン

1800: ヘリコプターのスプライトパターン

1890: プラットフォームのスプライトパターン

お絵かきプログラム

カーソルキーを使って画面に絵を描く「お絵かきプログラム」です。
ごく限られた機能しかありませんが、みなさんの手で改良してみてください。



※プログラムを止めるときは、CTRL を押しながら、STOP を押してください。

キーの機能

・色指定（透明と黒を含む16色指定可能）‥‥数字キーおよびA～Fの文字キー
・カーソルの移動（８方向）‥‥‥‥‥‥‥‥カーソルキー
・直線を描くモードと描かないモードの切替‥ SPACE
・同じ色の線で囲まれた部分に色を塗る‥‥‥ P
・画面を初期化する‥‥‥‥‥‥‥‥‥‥‥‥ DEL

使いかた

画面下部の色見本を参考に色を指定し、画面に表示されるカーソルを動かすことにより、図形を描きます。
カーソルの色と形が、現在のモードと色指定をあらわします。
・×形のカーソル‥‥‥‥‥‥‥‥‥‥‥‥‥線を描かないモード ┐
・十形のカーソル‥‥‥‥‥‥‥‥‥‥‥‥‥線を描くモード ┘ SPACE で切り替えます。
・カーソルの色‥‥‥‥‥‥‥‥‥‥‥‥‥‥現在の指定色
・点線のカーソル‥‥‥‥‥‥‥‥‥‥‥‥‥透明または黒の色指定（線の消去に使用）

ある色で完全に囲んだ部分の中にカーソルを入れ、同じ色を指定してから P を押すと、その部分を塗りつぶします。
囲んだ線と異なる色を指定して P を押したり、完全に囲んでいない部分で P を押すと、画面全体を塗りつぶしてしまいます。あやまって画面を塗りつぶしてしまったときは、DEL を押して画面を初期化し、初めから描き直してください。

```
1 'save"ｵｴｶｷ.BAS"
100 CALL ANK:SCREEN 5,0
110 OPEN "grp:" AS #1
120 '--- ｽﾌﾟﾗｲﾄ ﾉ ﾃｲｷﾞ
130 FOR I=0 TO 255:SPRITE$(I)
="":NEXT
140 FOR I=0 TO 31:PUT SPRITE
I,(0,217),,I:NEXT
150 FOR I=0 TO 3
160   B$=""
170   FOR N=0 TO 7
180     READ A$
190     B$=B$+CHR$(VAL("&B"+A
$))
200   NEXT N
210   SPRITE$(I)=B$
220 NEXT I
230 '--- ｶﾞﾒﾝｦ ｴｶﾞｸ
240 COLOR 15,0,0:CLS
250 YM=196
260 FOR C=0 TO 15
270   XM=C*16+4
280   PRESET (XM+1,YM):PRINT
#1,HEX$(C)
290   LINE(XM,YM+8)-STEP(7,7)
,C,BF
300 NEXT
310 LINE (0,0)-(255,195),15,B
320 X=128:Y=98:D=0:B=0:C=15:C
S=C
330 '--- ｵｴｶｷ ﾙｰﾁﾝ
340 CS=C:IF CS=<1 THEN CS=15
350 PUT SPRITE 0,(X,Y),CS,B+D

360 K$=INPUT$(1)
```

- 100: 英数・かなモード、スクリーン5を指定
- 110: 表示された画面を文字表示可能にする
- 130–140: 各スプライトパターンとスプライト面を初期化する
- 150–220: 4種類のカーソルのスプライトパターンを定義する
- 240–310: 初期画面を表示する
- 320: 各変数の値を初期設定する
- 350: カーソルのスプライトを表示する
- 360: キー入力を読む

```
370 IF K$=CHR$(&H7F) THEN 240

380 IF K$=CHR$(&H20) THEN D=1
-D:GOTO 330
390 IF "0"=<K$ AND K$=<"1" TH
EN C=VAL(K$)      :B=2:GOTO 33
0
400 IF "2"=<K$ AND K$=<"9" TH
EN C=VAL(K$)      :B=0:GOTO 33
0
410 IF "A"=<K$ AND K$=<"F" TH
EN C=VAL("&h"+K$):B=0:GOTO 33
0
420 IF "a"=<K$ AND K$=<"f" TH
EN C=VAL("&h"+K$):B=0:GOTO 33
0
430 IF K$="P" OR K$="p" THEN
GOSUB 520:GOTO 330
440 IF K$=CHR$(&H1C) AND    X
<250 THEN X=X+1:GOTO 490
450 IF K$=CHR$(&H1D) AND -3<X
     THEN X=X-1:GOTO 490
460 IF K$=CHR$(&H1E) AND -4<Y
     THEN Y=Y-1:GOTO 490
470 IF K$=CHR$(&H1F) AND    Y
<189 THEN Y=Y+1:GOTO 490
480 GOTO 330
490 '
500 IF D=1 THEN PSET (X+4,Y+5
),C
510 GOTO 330
520 '----- ﾍﾟｲﾝﾄ ｻﾌﾞﾙｰﾁﾝ
530 PRESET (X+4,Y+5):PAINT (X
+4,Y+5),C
540 RETURN
```

- 370: DEL が押されていたときの処理
- 380: SPACE が押されていたときの処理
- 390: 0 か 1 が押されていたときの処理
- 400: 2 ～ 9 が押されていたときの処理
- 410–420: A ～ F または a ～ f が押されていたときの処理
- 430: P が押されていたときの処理
- 440–470: カーソルキーが押されていたときの処理
- 480: 330行へ戻る
- 500: 線を描く処理
  描画モードなら指定色で点を描く
- 510: 330行へ戻る
- 530: 塗りつぶしの処理
  指定色で色を塗り、元の処理に戻る

```
550 '--- ｽﾌﾟﾗｲﾄ ﾊﾟﾀｰﾝ *
560 DATA 00000000
570 DATA 01000001
580 DATA 00100010
590 DATA 00010100
600 DATA 00001000
610 DATA 00010100
620 DATA 00100010
630 DATA 01000001
640 '
650 DATA 00000000
660 DATA 00001000
670 DATA 00001000
680 DATA 00001000
690 DATA 01111111
700 DATA 00001000
710 DATA 00001000
720 DATA 00001000
730 '
740 DATA 00000000
750 DATA 01000001
760 DATA 00000000
770 DATA 00010100
780 DATA 00000000
790 DATA 00010100
800 DATA 00000000
810 DATA 01000001
820 '
830 DATA 00000000
840 DATA 00001000
850 DATA 00000000
860 DATA 00001000
870 DATA 01010101
880 DATA 00001000
890 DATA 00000000
900 DATA 00001000
```

カーソルのスプライトパターンのデータ

住所録作成プログラム

ＢＡＳＩＣの漢字モードを使った、住所録作成プログラムです。これまでに説明していないフロッピーディスクのファイルを操作する命令を使いますが、かんたんな説明をつけています。



実行させせると画面にメニューが表示されますから、番号を入力して作業を選んでください。

※このプログラムを実行させせるときは、あらかじめＡドライブにフォーマット済みで書き込み可能なフロッピーディスクを入れておいてください。

```
1 'save"ｼﾞｭｳｼｮﾛｸ.BAS"'
100 '*** ｼﾞｭｳｼｮﾛｸ ﾉ ｻｸｾｲ ***
110 '* ｻｷﾞｮｳﾊﾞﾝｺﾞｳ ﾉ ｼﾃｲ *
120 CALL KANJI1:CLEAR 800
130 MAXFILES=2
140 CALL PALETTE:COLOR 15,4,
7:WIDTH 36:KEY OFF
150 LOCATE 6,1:PRINT "＊＊＊
 住所録 ＊＊＊"
160 LOCATE 8,3:PRINT "１－－
－新規作成"
170 LOCATE 8,4:PRINT "２－－
－追加"
180 LOCATE 8,5:PRINT "３－－
－全データ表示"
190 LOCATE 8,6:PRINT "４－－
－検索・表示"
200 LOCATE 8,7:PRINT "５－－
－削除"
```

- 110 ― 作業メニューの処理
- 120 ― 漢字モード１でメモリを初期化する
- 130 ― 同時に使用するファイル数の上限を宣言する
- 140 ― 画面の初期設定
- 150～200 ― 作業のメニューを表示する

```
210 LOCATE 8,8:PRINT "６－－
－終了"
220 LOCATE 6,10:INPUT "作業
番号の入力（１－６）";N1$
230 CALL KACNV(N$,N1$):N=VAL
(N$)
240 IF N<1 OR N>6 THEN 140
250 ON N GOTO 270,360,600,59
0,580,860
260 '* ｼﾝｷｻｸｾｲ *
270 OPEN "JYUUSHO.FAT" AS #1
280 FIELD #1,255 AS FT$
290 FF$=""
300 FOR F=1 TO 255
310 FF$=FF$+"0"
320 NEXT F
330 LSET FT$=FF$:PUT #1,1
340 CLOSE #1
350 '* ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ *
360 OPEN "JYUUSHO.FAT" AS #1
370 FIELD #1,255 AS FT$
380 GET #1,1
390 OPEN "JYUUSHO.DAT" AS #2
 LEN=84
400 FIELD #2,16 AS HN$,28 AS
 HT$,40 AS HJ$
410 FOR C1=1 TO 255
420 F1$=MID$(FT$,C1,1):IF F1
$="1" THEN 530
430 MID$(FT$,C1,1)="1"
440 CALL CLS:NM$=""
450 LOCATE 2,2:INPUT "氏名";
NM$
460 IF NM$="" THEN 560
470 LOCATE 2,3:INPUT "Tel ";
TL$
```

- 作業番号の入力を求める
- 入力された数字を半角文字にし、数値に変換する
- 数値（作業番号）によってジャンプする
- ディスクに索引用のファイルを作り、ファイル番号１とする
- ファイル１の変数領域の 255バイトを「ＦＴ$」とする
- 変数ＦＦ$の値を 255個の「０」にする
- 変数領域ＦＴ$にＦＦ$の値を入れ、変数領域の値をファイルに書き込む
- ファイルを閉じる
- ファイル１として索引用のファイルを開く
- ファイル１の変数領域の255バイトをＦＴ$に
- 変数領域にファイル１のデータを読み込む
- 住所録用のファイルを作り、ファイル番号２とする
- ファイル２の変数領域を３つに分ける
- 索引（ＦＴ$）の中で空きマーク（０）を探し、最初に見付けた空きマークを「１」に書き換える
- 登録するデータの入力を求める
- 「氏名」の入力で ↵ だけが押されたら、 560行にジャンプする
- 各項目のデータの入力を求める

```
480 LOCATE 2,4:INPUT "住所";
J$
490 LOCATE 2,6:INPUT "登録 1
:Ok/0:NG";A$
500 IF A$="１" OR A$="1" THE
N 510 ELSE 450
510 LSET HN$=NM$:LSET HT$=TL
$:LSET HJ$=J$
520 PUT #1,1:PUT #2,C1
530 NEXT C1
540 PRINT "   DATA AREA OVER
FLOW  !! "
550 FOR C2=0 TO 1500:NEXT C2
560 CLOSE:GOTO 140
570 '* ﾋｮｳｼﾞ･ｹﾝｻｸ･ｻｸｼﾞｮ *
580 LOCATE 6,12:INPUT "削除
データ（氏名）";D1$:GOTO 600
590 LOCATE 6,12:INPUT "表示
データ（氏名）";D1$
600 CALL CLS
610 OPEN "JYUUSHO.FAT" AS #1
620 FIELD #1,255 AS FT$
630 GET #1,1
640 OPEN "JYUUSHO.DAT" AS #2
 LEN=84
650 FIELD #2,16 AS HN$,28 AS
 HT$,40 AS HJ$
660 FOR C1=1 TO 255
670 F1$=MID$(FT$,C1,1)
680 IF F1$="0" THEN 800
690 GET #2,C1
700 IF N=3 THEN 730
710 CALL KINSTR(A,HN$,D1$)
720 IF A=0 THEN 800
730 PRINT HN$:PRINT HT$:PRIN
T HJ$:PRINT
```

- このままファイルに登録してよいかどうか確かめ、ダメなら450行に戻る（490〜500）
- 入力されたデータをファイル２の変数領域に転送する（510）
- ２つの変数領域の値をそれぞれのファイルに書き込み、次のデータの入力を求める（520〜530）
- データを入力しすぎたときの表示（540）
- すべてのファイルを閉じ、メニューに戻る（560）
- 削除のときは氏名の入力を求め、600行へ進む（580）
- 検索と表示のときは氏名の入力を求め、次の行へ進む（590）
- 索引ファイルと住所録ファイルを開く処理（610〜650）
- 索引を調べ、登録されているデータを読み込む処理（660〜690）
- 全データ表示なら 730行へジャンプする（700）
- 読み込んだデータに入力された氏名があるかどうか調べ、なければ 800行へ（710〜720）
- 氏名を含むデータを画面に表示する（730）

```
740 IF N=5 THEN 770
750 IF INKEY$="" THEN 750
760 A$=INKEY$:GOTO 800
770 INPUT "削除 1:Ok/0:NG";A
$
780 IF A$="１" OR A$="1" THE
N MID$(FT$,C1,1)="0"
790 PUT #1,1:CLOSE:GOTO 140
800 NEXT C1
810 CLOSE
820 PRINT "        DATA END "
830 PRINT "    HIT ANY KEY !
"
840 IF INKEY$="" THEN 840
850 A$=INKEY$:GOTO 140
860 CALL CLS:END
```

- 740: 削除するときは 770行へジャンプする
- 760: 何かキーが押されたら 800行へ
- 770: 削除してよいか確認を求める
- 780: 削除してよければ、索引のそのデータにあたる部分を「０」にする
- 790: 索引ファイルを書き換え、両方のファイルを閉じて140行へ戻る
- 810〜850: すべてのデータを表示し終わったときの処理
- 860: 終了の処理

第1章

「MSX-DOS」の紹介



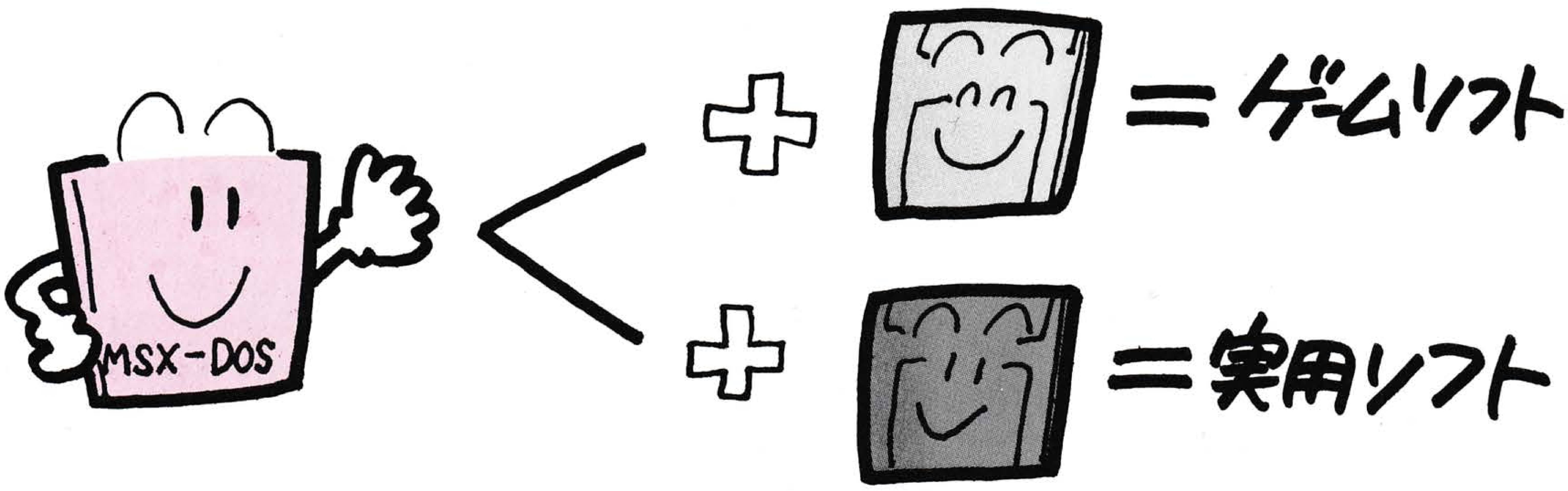
「MSX-DOS」って何だろう?

フロッピーディスクドライブ内蔵のＭＳＸ２＋パソコンおよび増設フロッピーディスクドライブ装置には、**「ＭＳＸ－ＤＯＳ」**と書かれたフロッピーディスクが付属しています。ＭＳＸは分かりますが、「ＤＯＳ」っていったい何なのでしょうか？

ＢＡＳＩＣはできるだけいろいろな用途に使えるようにたくさんの機能をもっているため、パソコンのメモリの半分以上をＢＡＳＩＣインタープリタが使っています。
しかし、パソコンを動かすためのプログラムは、何もＢＡＳＩＣでなければいけないわけではありません。実際、パソコンの付属ソフトや市販ソフトなどは、さまざまなプログラミング言語で書かれたあと、機械語に書き直した状態でフロッピーディスクに保存されて供給されています。

また、多くの市販ソフトは限られた目的のために作られていますから、そんなにいろいろな機能は必要ありません。それよりは自分自身のプログラムやデータの領域として使うほうが、メモリを有効に使えることになります。

そこで、フロッピーディスクに保存された機械語ソフトをより有効に使用できるように、フロッピーディスクドライブ内のファイルの操作に必要な機能だけを集めたものが**「ＤＯＳ」**（ドス：Disk Operating System）です。
「ＭＳＸ－ＤＯＳ」は、フロッピーディスクドライブと64キロバイト以上のメインメモリをもつすべてのＭＳＸパソコンのために作られたＤＯＳです。



MSX－DOSの用途

市販ソフトの中には、ＭＳＸ－ＤＯＳが組み込まれているものがたくさんあります。これらのソフトを使うときは、使う人がＭＳＸ－ＤＯＳのことを何も知らなくてもまったく問題ありません。

ここでは、ＭＳＸ－ＤＯＳの使いかたを知っていると便利な場合をあげます。

・**フロッピーディスクに保存したファイルを操作するとき**
ファイルの一覧を見たり、たくさんのファイルを他のディスクにコピーしたりするときには、ＢＡＳＩＣよりＭＳＸ－ＤＯＳを使うほうが便利です。

・**ＢＡＳＩＣ以外の言語でプログラムを自作するとき**
この場合、ＭＳＸ－ＤＯＳで動作する「コンパイラ」や「エディタ」などという機械語ソフトが必要になります。

MSX－DOSのスタート用のディスクの作りかた

一般にＭＳＸ－ＤＯＳは、パソコンやフロッピーディスクドライブの付属ディスクに保存されています。しかしＭＳＸ－ＤＯＳを使うときは、付属のディスクをそのまま使わず、コピーしたディスクを使うようにしましょう。
このようなディスクを**「システムディスク」**と呼びます。

初めてシステムディスクを作るときは、ＢＡＳＩＣを使って次のようにします。

① フォーマット済みの使用可能なフロッピーディスクを用意します。
② ＤＩＳＫ　ＢＡＳＩＣをスタートします。
③ ＢＡＳＩＣのＣＯＰＹ命令を使って、付属ディスクに保存されている次の２つのファイルを用意したディスクにコピーします。

```
MSXDOS.SYS
COMMAND.COM
```

これで用意したディスクがＭＳＸ－ＤＯＳのシステムディスクになりました。

DOS編

第2章

BASICとはひと味ちがうMSX-DOS。このシンプルさが快感です。

MSX-DOSの使いかた



MSX-DOSを使ってみよう

では、ＭＳＸ－ＤＯＳの使いかたをかんたんに説明します。

MSX-DOSのスタートのしかた

内蔵ソフト切り替えスイッチがあるパソコンでは、あらかじめ「切」にします。

① システムディスクをドライブＡに入れます。

② パソコンのリセットボタンを押すか、または電源スイッチを入れ直します。

③ 画面に下の文字が表示されます。これでＭＳＸ－ＤＯＳがスタートしています。

```
MSX-DOS version X.XX
Copyright 1984 by Microsoft

COMMAND version X.XX

A>█
```

MSX-DOSの基礎知識

「Ａ＞」

ＢＡＳＩＣのダイレクトモードでは、パソコンが命令の入力を待っているとき、画面に「Ｏｋ」と表示されました。
ＭＳＸ－ＤＯＳでこれにあたるのが、**「Ａ＞」（Ａプロンプト）**です。
左端の「Ａ」は、ドライブ名を省略して命令したときにＭＳＸ－ＤＯＳが操作するフロッピーディスクドライブ（この場合、Ａドライブ）をあらわします。

「カレントドライブ」

ドライブ名を省略して操作できるドライブを**「カレントドライブ」**と呼びます。
ＢＡＳＩＣではドライブ名を省略できるドライブはＡドライブに限られましたが、ＭＳＸ－ＤＯＳでは、カレントドライブを変更できます。
ためしに「Ｂ：」と入力してみましょう。「Ａ＞」が「Ｂ＞」に変わります。
カレントドライブがＢドライブに変更されました。
こんどは、「Ａ：」と入力してみましょう。再び「Ａ＞」が表示されます。

「2ドライブシミュレーション」

内蔵または接続しているドライブが1台だけのときでも、同じドライブをA・B、2つのドライブとして使うことができます。ただし、この場合には、カレントドライブの変更後最初にドライブに対する操作を行う命令を入力したとき、およびカレントドライブでないほうのドライブ名を指定したときに下のようなメッセージが画面に表示されてドライブ内のディスクの入れ替えを求めてきます。

```
Insert diskette for drive B: (またはA:)
and strike a key when ready■
```

⏎を押して初めて、ドライブ内のディスクを指定されたドライブのディスクとみなして命令を実行します。
これは**「2ドライブシミュレーション」**と呼ばれる機能で、ドライブ1台でファイルをコピーするときなどに使用します。

「コマンドライン」

ＢＡＳＩＣでは「Ok」の下にカーソルが表示されましたが、ＭＳＸ－ＤＯＳでは「>」の右にカーソルが表示されます。

A>■

現在カーソルが表示されている「>」がある行を**「コマンドライン」**と呼びます。
ＭＳＸ－ＤＯＳでは、命令の入力はすべて「コマンドライン」で行います。

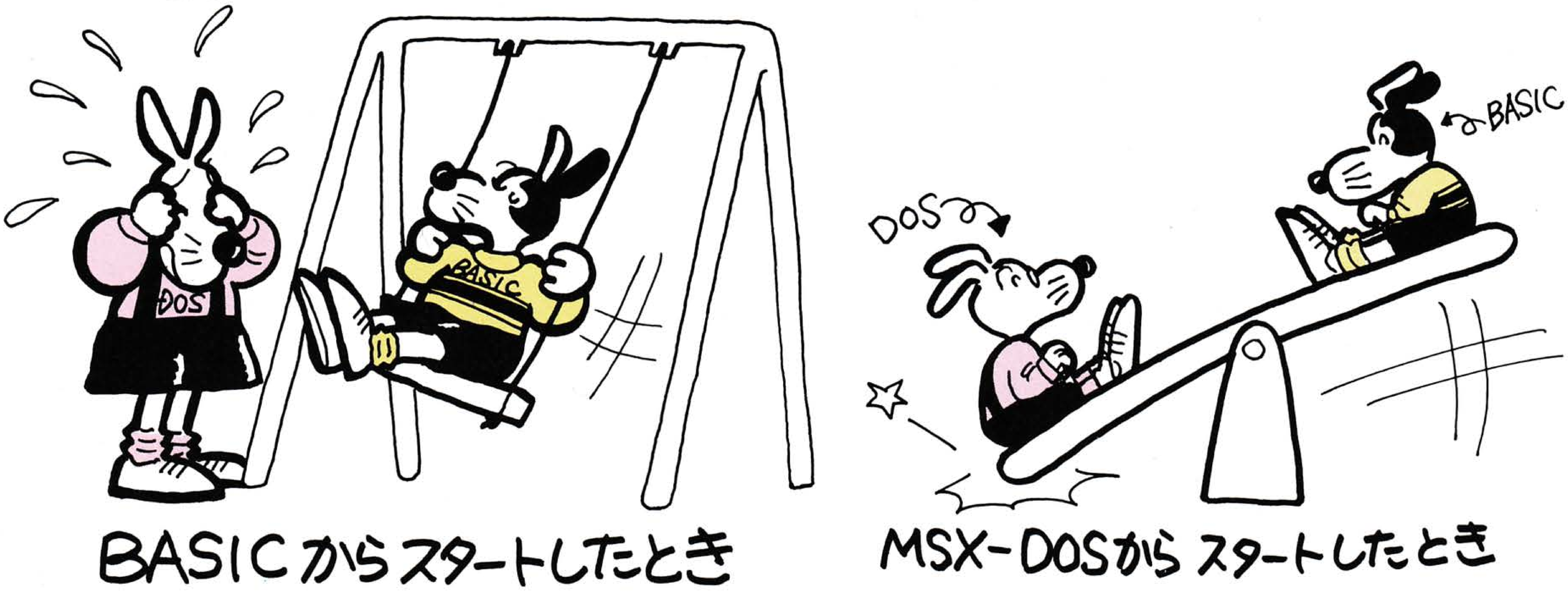
ＭＳＸ－ＤＯＳからＢＡＳＩＣをスタートさせる　（ＢＡＳＩＣ）

ＭＳＸ－ＤＯＳには、**「ＢＡＳＩＣ」**という命令があります。これは、ＭＳＸ－ＤＯＳからＤＩＳＫ　ＢＡＳＩＣをスタートさせる命令です。

ＢＡＳＩＣ命令でＭＳＸ－ＤＯＳからＤＩＳＫ　ＢＡＳＩＣをスタートさせたときは、**「ＣＡＬＬ　ＳＹＳＴＥＭ」**（コール・システム）という命令を実行させることにより、再びＭＳＸ－ＤＯＳに戻ることができます。

（このときは、Ａドライブにシステムディスクが入っていることが必要です）

パソコンをＢＡＳＩＣからスタートさせたときは、ＣＡＬＬ　ＳＹＳＴＥＭ命令を使えません。



ファイルの一覧を表示する　（ＤＩＲ）

ＢＡＳＩＣで、フロッピーディスクに保存されているファイルの一覧を見るには、ＦＩＬＥＳ命令を使いました。ＭＳＸ－ＤＯＳでそれにあたるのが**「ＤＩＲ」**（ディレクトリ）命令です。

ＤＩＲ　ドライブ名　ファイルスペック

例）
```
ＤＩＲ
ＤＩＲ　Ａ：
ＤＩＲ　ＣＯＭＭＡＮＤ．ＣＯＭ
ＤＩＲ　Ａ：ＭＳＸＤＯＳ．ＳＹＳ
```

ＭＳＸ－ＤＯＳのファイルの一覧は、下のような形式になります。

・40桁表示のとき

ＭＳＸＤＯＳ	ＳＹＳ	２４３２	８５－０８－２３
（ファイル名）	（拡張子）	（ファイル容量）	（保存された日付）

・80桁表示のとき

MSXDOS　SYS　2432 85-08-23　9:29p　（保存された時刻）

ファイル一覧の下には、保存されているファイルの総数と、ディスクの残り容量が表示されます。

```
     2 files ―――――― ファイルの総数
719872 bytes free ―― 残り容量
```

命令の入力のしかた

ＤＩＲ命令の例でも分かるように、ＭＳＸ－ＤＯＳではＢＡＳＩＣと命令の入力のしかたが少し異なります。

・ファイルスペックは「″」で囲まずに入力します。
・命令とファイルスペックなど、入力する各項目のあいだには、必ず１文字以上の空白を入れます。ただし、ドライブ名とファイル名、ファイル名と「．」を含めた拡張子のあいだ（つまりファイルスペックの中）には空白を入れてはいけません。
・１つのコマンドラインで２つ以上の命令を入力することはできません。
・ＢＡＳＩＣのプログラムモードにあたるものはありませんが、そのかわり、決まった手順で行う命令を登録する**「バッチファイル」**というファイルを作ることができます。バッチファイルはコマンドラインにファイルスペックを入力するだけで実行されます。

ファイルをコピーする　（ＣＯＰＹ）

フロッピーディスクに保存されているファイルを他のディスクにコピーするときは、次のようにします。（例として、「ＣＯＭＭＡＮＤ．ＣＯＭ」をコピーします）

●ドライブ１台でコピーするとき　（ドライブ２台でコピーするときは、144ページ）

① フォーマット済みの書き込み可能なフロッピーディスクを用意します。
② ＭＳＸ－ＤＯＳをスタートします。
③ コピーする元のディスクのライトプロテクトタブを「書き込み禁止」の位置にセットし、ドライブに入れます。
④ コマンドラインに次のように入力します。

```
A>COPY COMMAND.COM B:
```

⑤ アクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯し、「ＣＯＭＭＡＮＤ．ＣＯＭ」がパソコンのメモリに呼び出されます。

⑥ 次のように画面に表示されます。

```
Insert diskette for drive B:
and strike a key when ready■
```

⑦ アクセスランプ（またはＩＮ　ＵＳＥランプ）が消えてから、コピー元のディスクと用意したディスクを入れ替え、↵を押します。
⑧ アクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯し、ＣＯＭＭＡＮＤ．ＣＯＭが保存されます。
⑨ 画面に次のように表示されます。

```
Insert diskette for drive A:
and strike a key when ready■
```

⑩ ランプが消えてから、再びディスクを入れ替え、↵を押します。
⑪ ファイルが大きすぎて一度にコピーできないときは、⑥～⑩の操作を繰り返します。

⑪ コピーが終わると、画面に下のように表示されます。

```
        1 file copied
A>■
```

●ドライブ２台でコピーするとき

メモ： ドライブ１台内蔵のパソコンにドライブをもう１台増設したときは、増設したドライブが「ドライブＡ」、内蔵ドライブが「ドライブＣ」になります。
：ドライブ２台内蔵のパソコンでは、片方が「ドライブＡ」、もう一方が「ドライブＢ」になります。どちらがＡ，Ｂかは、本体に表示されています。
：ドライブを内蔵していないパソコンにドライブ２台を接続したときは、スロット１に接続したほうが「ドライブＡ」、スロット２が「ドライブＣ」になります。

① フォーマット済みの書き込み可能なフロッピーディスクを用意します。
② ＭＳＸ－ＤＯＳをスタートします。

③ コピーする元のディスクのライトプロテクトタブを使用禁止の位置にセットし、ドライブAに入れます。
④ 用意したディスクをドライブC（またはドライブB）に入れます。
⑤ コマンドラインに次のように入力します。

A>COPY COMMAND.COM C:
ドライブを増設しているとき

または、

A>COPY COMMAND.COM B:
ドライブ2台内蔵のとき

⑥ ドライブAのアクセスランプ（またはIN USEランプ）が点灯し、COMMAND．COMがパソコンのメモリに呼び出されます。
⑦ ドライブAのアクセスランプ（またはIN USEランプ）が消え、かわってドライブC（またはB）のアクセスランプ（またはIN USEランプ）が点灯してCOMMAND．COMが保存されます。
⑧ ファイルが大きすぎて一度にコピーできないときは、⑥と⑦の動作が繰り返されます。
⑨ コピーが終わると、画面に下のように表示されます。

1 file copied
A>■

ディスクをコピーする

1枚のフロッピーディスクに保存されているすべてのファイルを他のディスクにコピーするときは、次のようにします。

●ドライブ1台でコピーするとき　（ドライブ2台でコピーするときは146ページ）

① フォーマット済みの書き込み可能なフロッピーディスクを用意します。
② MSX－DOSをスタートします。
③ コピーする元のディスクのライトプロテクトタブを「書き込み禁止」の位置にセットし、ドライブに入れます。
④ コマンドラインに次のように入力します。

A>COPY *.* B:

⑤ アクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯し、メモリの空き容量が許すかぎりのファイルが連続して呼び出されます。
画面には呼び出したファイルのファイル名が次々に表示されます。

⑥ 画面に次のように表示されます。

```
Insert diskette for drive B:
and strike a key when ready■
```

⑦ アクセスランプ（またはＩＮ　ＵＳＥランプ）が消えてから、コピー元のディスクと用意したディスクを入れ替え、↵を押します。

⑧ アクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯し、メモリに呼び出されたファイルが次々に保存されます。

⑨ 画面に次のように表示されます。

```
Insert diskette for drive A:
and strike a key when ready■
```

⑩ ランプが消えてから、再びディスクを入れ替え、↵を押します。

⑪ 画面に下のように表示されるまで、⑤～⑩の操作を繰り返します。

```
        ×× files copied
A>■
```

●ドライブ２台でコピーするとき

メモ： ドライブ１台内蔵のパソコンにドライブをもう１台増設したときは、増設したドライブが「ドライブＡ」、内蔵ドライブが「ドライブＣ」になります。
：ドライブ２台内蔵のパソコンでは、片方が「ドライブＡ」、もう一方が「ドライブＢ」になります。どちらがＡ，Ｂかは、本体に表示されています。
：ドライブを内蔵していないパソコンにドライブ２台を接続したときは、スロット１に接続したほうが「ドライブＡ」、スロット２が「ドライブＣ」になります。

① フォーマット済みの書き込み可能なフロッピーディスクを用意します。
② MSX－DOSをスタートします。
③ コピーする元のディスクのライトプロテクトタブを「書き込み禁止」の位置にセットし、ドライブAに入れます。
④ 用意したディスクをドライブC（またはドライブB）に入れます。
⑤ コマンドラインに次のように入力します。

A＞COPY ＊.＊ C： ——— ドライブを増設しているとき

または、

A＞COPY ＊.＊ B： ——— ドライブ2台内蔵のとき

⑥ ドライブAのアクセスランプ（またはIN USEランプ）が点灯し、メモリの空き容量が許すかぎりのファイルが連続して呼び出されます。
⑦ 画面には呼び出したファイルのファイル名が次々に表示されます。
ドライブAのアクセスランプ（またはIN USEランプ）が消え、かわってドライブC（またはB）のアクセスランプ（またはIN USEランプ）が点灯してメモリに呼び出されたファイルが次々に保存されます。
⑧ すべてのファイルのコピーが終わるまで、⑥と⑦の動作が繰り返されます。
コピーが終わると、画面に下のように表示されます。

```
        ×× files copied
A>■
```

ワイルドカード

ディスクのコピーで、入力した**「＊」**（アスタリスク）は、ファイルスペックの代わりに入力したときは、文字数の決まっていない不特定の文字列をあらわし**「ワイルドカード」**と呼ばれます。
ワイルドカードに使える文字には「＊」の他にも**「？」**（クエスション）があり、ファイルスペックの中で使うときには、不特定の1文字をあらわします。
例えば「＊.＊」はすべてのファイルをあらわし、「PROGRAM?」は、「PROGRAM」の後ろに何か1文字続くファイル名をもつすべてのファイルをあらわします。

こんなメッセージが表示されたときは

ＭＳＸ－ＤＯＳは、ＢＡＳＩＣとは異なるエラーメッセージを表示します。
その中で、次のような表示が出たときは、指示を入力することが必要です。

```
Abort, Retry, Ignore?■
```

エラーが発生した操作（またはバッチファイルの実行）を、

・中止するとき　…………　[A]キーを押す。
・再実行するとき　…………　[R]キーを押す。
・無視して先に進むとき…………　[I]キーを押す。

※ [I]キーを押して処理を続けると、ファイルやディスクを壊すことがあります。
通常は使用しないでください。

資　料　編

漢字BASICの規則概説

これからのページでは、「漢字ＢＡＳＩＣ」（およびＤＩＳＫ　ＢＡＳＩＣ）の一般的規則の概略を説明しています。必要なときにお読みください。

※ＢＡＳＩＣは初めてという人は、いきなりこの「規則概説」を読んでも理解しにくいかもしれません。まずこの本の前のほう（「ＢＡＳＩＣ編」）で説明されている規則を理解するようにしてください。特に複雑なプログラム以外は、それだけで十分に作れます。ＢＡＳＩＣでのプログラム作成になれてくれば、ここに書いてあることも自然に理解できるようになるでしょう。

漢字BASICで扱えるデータ(定数・変数)

分類		内容	型宣言文字（＊1）
文字型		255文字以内の文字列	$ 例　A$ =″MOJI″
数値型	整　数	−32768～＋32767の整数 （小数点以下の値は切り捨て） 表示形式（＊4）	% 例 12.99%→12 A%＝123
	単精度実数	有効数字６桁の精度の実数 指数範囲は 10^{-64} ～ 10^{+62} 指数定数の表記（＊2）	！　（＊3） 例 12345678! →12345700 A！＝1.2345
	倍精度実数	有効数字14桁の精度の実数 指数範囲は 10^{-64} ～ 10^{+62} 指数定数の表記（＊2）	＃　（＊3） 例 1.23456789# A＃＝98.765432

＊１）**型宣言文字**　変数名や定数の後ろに付けることにより、そのデータの型を宣言（定義）する文字。型宣言文字を付けていない定数および変数（ＤＥＦＳＴＲ，ＤＥＦＩＮＴ，ＤＥＦＳＮＧ，ＤＥＦＤＢＬ文により定義されているものを除く）は、倍精度型実数とみなされます。

＊２）**指数定数の表記**　**（数値）Ｅ±ｎ**　ｎは指数（−６４～＋６２）
例　１２３４５６．７８＝１．２３４５６７８Ｅ＋５　０．００００３４＝３．４Ｅ−６

＊３）型宣言による**有効桁を超える数値**は、有効桁の次の桁で四捨五入されます。

例

定義式の例	取り扱われる数値
Ａ！＝１２３４．５６７８９０１２３４	１２３４．５７
Ａ#＝１２３４．５６７８９０１２３４５６	１２３４．５６７８９０１２３５

＊4） **整数の表示形式**

整数は2バイトで、各ビットに下記のような「重み」を付けたビット配列であらわします。

2バイト目（上位）								1バイト目（下位）							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

16進形式　先頭に&Hを付けた0～9，A～Fの並び。2進形式のビット配列を4ビットずつ区切ってあらわす。

（例　&B1111101001011101=&HFA5D）

10進形式の　10，11，12，13，14，15　をそれぞれ　A，B，C，D，E，F　に対応させてあらわす。

変　数

変数とは、プログラム中で使う文字列や数値をしまっておくことができる「入れ物」のことで、各変数にはそれぞれメモリの特定の領域が割り当てられます。変数には区別のため、それぞれ変数名を付けます。

変数名
- 2文字までの文字。（1文字目は英字、2文字目は英字または数字）3文字目以降は無視。ただし、型宣言文字は変数名の文字数として数えない。
- 英字の大文字、小文字は区別しない。（ABとabは同じ変数となる）
- 予約語を含まないこと。（⇨ 190ページ「予約語表」）

配列変数

配列変数とは添え字により区別されたいくつかの要素から構成された変数です。

例　AA（L，M，N）

- AA：変数名
- L，M，N：各次元の要素数（各要素は０から始まる）
- （L，M，N）：配列の次元数（上の例は３次元）

配列変数は、使用する前にＤＩＭ文で配列の大きさ（次元数と各次元の要素数の上限）を宣言して使用します。

例　DIM　A（10，20，5）

ただし、３次元までの配列変数は宣言なしでも使用できます。この場合、各次元の要素数の上限は、自動的に10まで（０を含め11個）に設定されます。

演算子・関係演算子・論理演算子

ＢＡＳＩＣで使用する演算子（計算に用いる記号）は、次のようになっています。

算術演算子	指数	負号	×，÷	整数除算	剰余	＋，－
	＜	－	＊，／	￥	MOD	＋，－

関係演算子	等号	より大	より小	以上	以下	不等号
	＝	＜	＞	＜＝ ＝＜	＞＝ ＝＞	＜＞ ＞＜

※関係演算子は２つの数値または文字列を比べるのに用います。演算子がしめす関係が、正しければ結果は真（－１）に、正しくなければ偽（０）になります。

論理演算子	否定	論理積	論理和	排他的論理和	包含	同値
	NOT	AND	OR	XOR	IMP	EQV

※論理演算子は複数の条件を調べたり、データをビット操作したりするときなどに用います。

例） 4ビットのデータ2個のビット操作例

入力条件	X	0	1	0	1
	Y	0	0	1	1
NOT X		1	0	1	0
X AND Y		0	0	0	1
X OR Y		0	1	1	1
X XOR Y		0	1	1	0
X IMP Y		1	0	1	1
X EQV Y		1	0	0	1

操作した結果（NOT X～X EQV Y）

演算の優先順位

・ひとつの式の中では、各表の左側の演算子から先に計算される。
・表が異なる演算子のあいだでは、算術演算子・関係演算子・論理演算子の順に計算される。
・式の中に（　）がある場合は、内側の（　）の中から先に計算される。

漢字BASICの基本モード

漢字ＢＡＳＩＣは次のような構成になっています。

漢字ＢＡＳＩＣ

英数・かなモード	**漢字モード**
テキスト画面 （スクリーンモード０，１）	**漢字テキスト画面** （スクリーンモード０，１）
グラフィック画面 （スクリーンモード２～12）	**漢字グラフィック画面** （スクリーンモード２～12）

２つの基本モードは、次の命令で切り替えます。

英数・かなモード	→	漢字モード	……	ＣＡＬＬ　ＫＡＮＪＩ
漢字モード	→	英数・かなモード	……	ＣＡＬＬ　ＡＮＫ

英数・かなモードの特徴

・従来のＭＳＸ２　ＢＡＳＩＣで作成したプログラムが変更なしで実行できる。

漢字モードの特徴

・漢字テキスト画面に漢字などの全角文字を表示できる。
・漢字グラフィック画面に漢字を表示するときでも、ＬＯＣＡＴＥ文やＰＲＩＮＴ文が使用できる。
・かな漢字変換機能がつき、キーボードからかなを入力して漢字に変換することができる。
・ＢＡＳＩＣのプログラム中の文字列やＲＥＭ文などに漢字を直接使用できる。
・漢字などの全角文字の文字列をあつかう命令や関数が追加されている。

画面モード

スクリーンNo.	モード	最大表示桁数または解像度	表示色数	色指定	スプライト
0	テキストモード	80桁×24行	16パレット中の2色	前景色 背景色 周辺色	表示できない
1		32桁×24行			8×8ドット 1ラインに4個まで 同時表示可能
2	高解像度グラフィックモード	256×192ドット	16パレット	8ドットを1単位とする領域ごとに2色	
3	マルチカラーモード	64×48ブロック		1ブロック毎に指定	
4	高解像度グラフィックモード	256×192ドット		8ドットを1単位とする領域ごとに2色	8×8ドットまたは 16×16ドット 1ラインに8個まで 同時表示可能
5	ビットマップグラフィックモード	256×212ドット		1ドット毎に指定	
6		512×212ドット	4パレット		
7			16パレット		
8		256×212ドット	256色（固定）		
10	自然画表示モード		パレット・YJK混在 16パレット YJK表示時、 12，499色	パレット表示は、 1ドット毎に指定 YJK表示は、連続した4ドット毎に指定	
11					
12			YJK表示のみ 19，268色	連続した4ドット毎に指定	

※パレットは、赤緑青各8段階の組み合わせからなる 512色中よりパレットごとに色指定することができる。
※1ブロックは、4×4ドットで構成。
※自然画は、カラー写真などを特殊な装置を使って画像データに変換した絵のこと。
※ＹＪＫは、従来のＲＧＢ方式とは異なる自然画表示のための色表現方式。（⇨ 159ページ）
ＢＡＳＩＣの命令では直接使用できない。

英数・かなモードの文字表示

画面モード	ANKサイズ	表示命令	位置指定命令	漢字サイズ	表示命令
スクリーン0	6×8ドット	PRINT	LOCATE	表示できない	
スクリーン1	8×8ドット	PRINT	LOCATE	表示できない	—
スクリーン2	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン3	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン4	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン5	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン6	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン7	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン8	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン10	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン11	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI
スクリーン12	8×8ドット	OPEN, PRINT#	最終参照点	16×16ドット	PUT KANJI

※「ANK」は、アルファベット・数字・カタカナ・ひらがな・グラフィック文字およびそれらの記号などの、キーボードから直接入力できる文字の総称。

※スクリーン11～12は、表示色の設定方法が他の画面と異なるため、文字表示に適さない。

※最終参照点は、最後に実行したグラフィック関係の命令で指定した位置座標のこと。通常は、PRESET命令などを使って設定する。

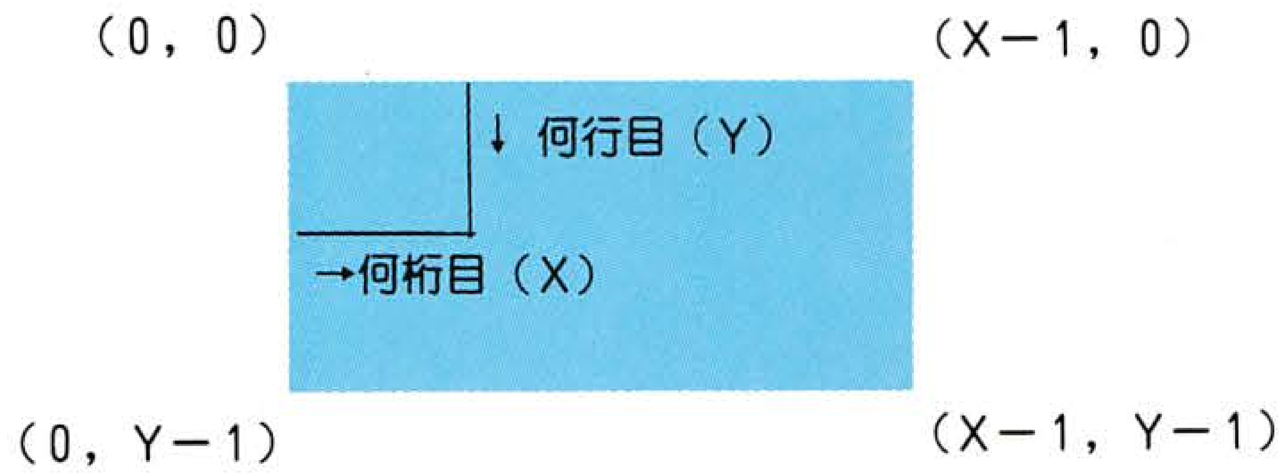
スクリーン0の文字表示

・スクリーン0で1行が41桁以上のときは、画面のヨコ方向のドット密度が2倍になるため、文字の幅が半分になる。

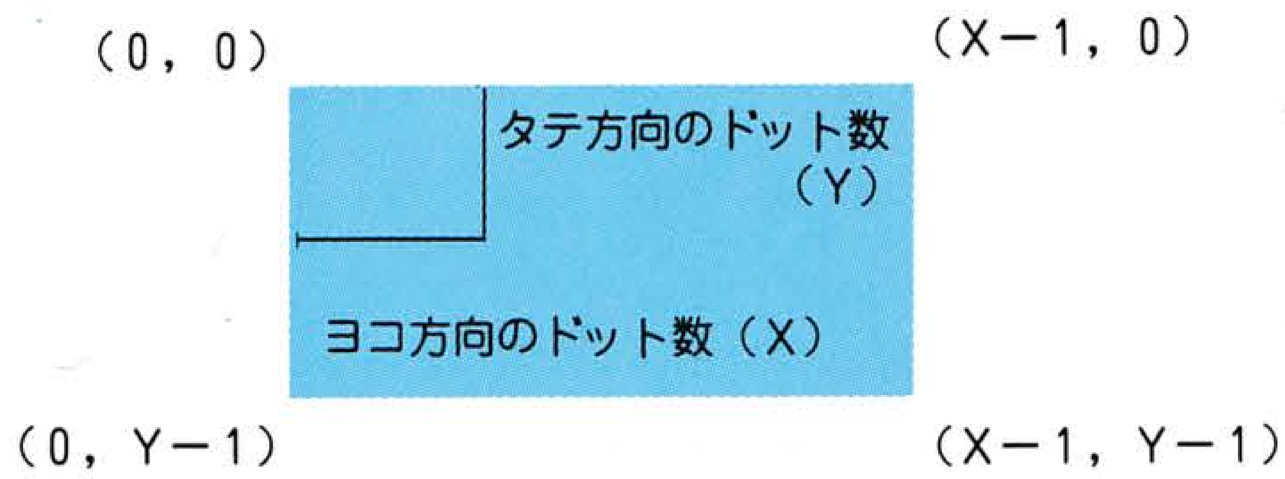
・スクリーン0では文字の表示パターンの右2列分のドットをカットして表示するため、ひらがなやグラフィック文字の一部は文字の右端が欠けて表示される。

文字表示の位置座標

・ＬＯＣＡＴＥ文による文字表示位置の指定は文字の桁・行単位。画面の左上角を（０，０）とする。



・ＰＲＥＳＥＴ文による文字表示位置の指定はドット単位。画面の左上角を（０，０）とし、指定した位置を文字の表示パターンの左上角とする。



・ＰＵＴ　ＫＡＮＪＩ文は表示文字の指定と同時に位置の指定をすることができる。
位置の指定方法はＰＲＥＳＥＴ文と同じ。

漢字モードの文字表示

漢字モードは、文字表示の方法により、次の４種類に分かれています。

漢字モード	全角文字サイズ	半角文字サイズ	最大表示文字数	インターレース表示
ＫＡＮＪＩ０	16×16ドット	8×16ドット	32桁×13行	しない
ＫＡＮＪＩ１	12×16ドット	6×16ドット	40桁×13行	しない
ＫＡＮＪＩ２	16×16ドット	8×16ドット	32桁×24行	する
ＫＡＮＪＩ３	12×16ドット	6×16ドット	40桁×24行	する

※最大表示文字数は、漢字テキスト画面でＷＩＤＴＨ命令により最大桁数を指定し、ＫＥＹ　ＯＦＦ命令を実行させた状態での全角文字の表示数をあらわします。
半角文字の表示桁数は全角文字の２倍になります。
※インターレース表示は、一般の家庭用テレビではちらつきが大きく、実用的ではありません。

漢字モードでのＷＩＤＴＨ命令

・漢字テキスト画面でのみ有効です。漢字グラフィック画面で実行させるとエラーになります。
・指定する桁数は半角文字を基準にします。
・指定できる桁数の範囲は 26～ 64(または80)になります。
・漢字グラフィック画面では、同じ漢字モードの漢字テキスト画面で設定可能な最大の桁数・行数に自動的に設定されます。
・漢字テキスト画面の横方向の表示ドット数は、漢字モードとＷＩＤＴＨ命令によって変化します

ＫＡＮＪＩ０またはＫＡＮＪＩ２の場合

ＷＩＤＴＨ２６～３２のとき　……　256ドット
ＷＩＤＴＨ３３～６４のとき　……　512ドット

ＫＡＮＪＩ１またはＫＡＮＪＩ３の場合

ＷＩＤＴＨ２６～４０のとき　……　256ドット
ＷＩＤＴＨ４１～８０のとき　……　512ドット

漢字テキスト画面での文字の表示色

英数・かなモードでは、文字の表示色は前景色に固定されているため、ＣＯＬＯＲ命令で前景色の指定を変えると、テキスト画面に表示されているすべての文字の色が変わります。
漢字テキスト画面では、文字の表示色は「その文字を表示したときの前景色」になります。
前景色の指定を変えながら表示することにより、１文字ごとに表示色を変えることができます。

グラフィック画面

グラフィック画面の１ドット（画素）に対応するＶＲＡＭと表示色との関係は、次ページの図のようになっています。（図中、ーはＶＲＡＭ１バイトの各ビットをあらわします）

スクリーンNo.	VRAM 1バイト	色　数
2 4	（1ビット／1ドット）	8ドットを1単位とする領域ごとに1バイトのカラーテーブルを割り当て、色指定。（各領域ごとに16パレット中より2色の表示が可能）＊1）
3	（4ビット／1ブロック）	各ブロックごとに16パレット
5	（4ビット／1ドット）	各ドットごとに16パレット
6	（2ビット／1ドット）	各ドットごとに4パレット
7	（4ビット／1ドット）	各ドットごとに16パレット
8	（8ビット／1ドット）	各ドットごとに 256色
10 ＊2） 11 ＊2）	D1 Y1 A1 KL D2 Y2 A2 KH D3 Y3 A3 JL D4 Y4 A4 JH （4バイト／4ドット）	Aのビットが0の1バイトに対応する4ドットは、K，J（各6ビット）により4ドット単位で色指定し、Y（4ビット）により各ドットごとの輝度を指定 （YJK方式） Aのビットが1の1バイトに対応する1ドットは、Yのデータによる16パレット （RGB方式）
12	D1 Y1 KL D2 Y2 KH D3 Y3 JL D4 Y4 JH （4バイト／4ドット）	YJK方式 K、J（各6ビット）により、4ドット単位で色指定 Y（5ビット）により、各ドットごとに輝度を指定

＊1）8ドットを単位とする領域に対して3番目の色を指定すると、先に指定した色が変化します。

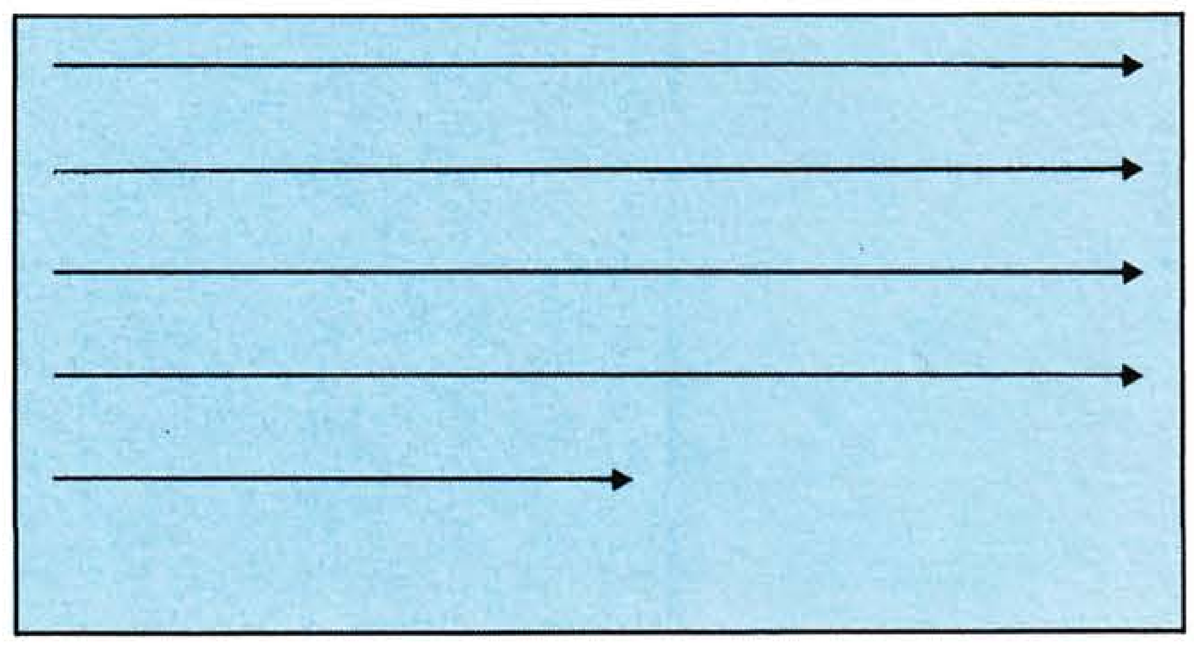
＊2）スクリーン10と11の異なる点

スクリーン10：描画の際に指定されたカラーコードを、そのままパレットの番号とみなします。（カラーコードのデータは上位4ビット（Yの部分）にセットする）
各ドットに対応する1バイトのAは、自動的に1にします。

スクリーン11：ＢＡＳＩＣの描画命令（ＬＩＮＥやＰＡＩＮＴなど）を実行しても、Ａは１に設定しません。したがって、描画の際に指定されたカラーコード（０～255）を、単なる１バイト分のデータとみなしてＹＪＫ方式で表示します。

参考　ＶＲＡＭの画面との対応

右の図のように画面の最上段の走査線の左端より右端、２段目の走査線の左端より右端‥‥の順で、各ドットのカラーコードが上の表にしめした形で格納されます。



色指定

表示の色は、**カラーコード**により指定します。あらかじめカラーコードに割り当てられている色を右の表に示します。（色の成分は、**R**（赤）、**G**（緑）、**B**（青）の各成分をそれぞれ０～７で示したものです。）

カラーコードと色との関係は固定されたものではなく、各カラーコードごとに 512色中から１色を指定して色の割り当てを変えることができます。このような機能を**「パレット」**と呼びます。

色の割り当てを変更する命令がＣＯＬＯＲ＝文です。

例） カラーコード６（暗い赤）を赤にする。

```
COLOR＝(6，7，1，1)
        |  赤  緑  青
カラーコード   |
           色の成分
```

コード No.	最初の表示色	色の成分 赤	緑	青
0	透明	0	0	0
1	黒	0	0	0
2	緑	1	6	1
3	明るい緑	3	7	3
4	暗い青	1	1	7
5	明るい青	2	3	7
6	暗い赤	5	1	1
7	水色	2	6	7
8	赤	7	1	1
9	明るい赤	7	3	3
10	黄	6	6	1
11	明るい黄	6	6	4
12	暗い緑	1	4	1
13	紫	6	2	5
14	灰	5	5	5
15	白	7	7	7

◆カラーコード０の色を変更するときは、あらかじめ次の命令を実行させてください。

VDP(9)=VDP(9) OR &H20　　　（元に戻すとき）　VDP(9)=VDP(9) AND &HDF

グラフィック関係の補足説明

漢字モードで使用できない命令

次の命令は、英数・かなモードでは使用できますが、漢字モードで実行させようとするとエラーになります。

CLS ………… 漢字テキスト画面で実行させると、エラーになります。
漢字グラフィック画面では、エラーにならずに使用できます。
※漢字テキスト画面で画面表示を消去するときは、CALL　CLS命令を使用します。（⇨ 174ページ）

COLOR= …… COLOR文には、いくつかの機能がありますが、各カラーコードに割り当てられている色を変更する機能（パレット機能）は、漢字テキスト画面では使用できません。
例）　COLOR=（8，1，6，1）などを実行するとエラーになる
漢字グラフィック画面では、支障なく使用できます。
その他の機能は、漢字テキスト画面でも使用できます。
※漢字テキスト画面でパレットの設定を変更したいときは、CALL　PALETTE命令を使用します。（⇨ 174ページ）

新たに追加された命令

グラフィック関係で新たに追加された命令で、本文中で説明していないものには、次の命令があります。

SET　SCROLL（セット・スクロール） …… 画面をタテおよびヨコ方向にスクロールさせます。スクリーン5～8では、2ページ連続でヨコ方向にスクロールできます。

```
SET SCROLL 128,100,1,0
```

（ヨコのスクロール量）128
（タテのスクロール量）100
（目隠し指定）1
（次ページ指定）0

ヨコのスクロール量は0～511の数値
タテのスクロール量は0～255の数値
目隠し指定は0か1。1の場合、水平スクロールしたとき左端の8ドットを周辺色で隠す。0の場合は隠さない。

次ページ指定は0か1。0のときは水平スクロール時にそのページ内で画面の両端がつながってスクロールする。1のときは2ページ連続で水平スクロールする。ただし表示ページは奇数に設定する。

資料編

ファイル

パソコンに入力したプログラムやデータは、ファイルとしてフロッピーディスクやカセットテープに保存し、必要なときにパソコンのメモリの中に呼び出して再実行させることができます。

ファイルスペック

ファイルスペックはファイルの入出力の際に、対象となる入出力機器やファイルの名前を指定するためのものです。ファイルスペックは次のような形であらわします。

> **"デバイス名：ファイル名．拡張子"**

ファイルスペックは必ず「"」で囲みます。
デバイス名とファイル名のあいだは「：」（コロン）
ファイル名と拡張子のあいだは「．」（ピリオド）で区切ります。

デバイス名・ファイル名・拡張子は、それぞれ省略できる場合があります。

入出力機器	デバイス名 ＊1)	ファイル名 ＊2)	拡張子 （⇨ 38ページ）
カセットテープ	CAS：	6文字以内 ・英字の大文字と小文字を区別する	なし （使用できない）
フロッピーディスク	A：～H：	8文字以内 ・英字の大文字、小文字を区別しない 例）"abc" と "ABC" は同じファイル名	3文字以内 *2)
RAMディスク	MEM：		
テキスト画面	CRT：	6文字以内 （省略可能）	なし （使用できない）
グラフィック画面	GRP：		
プリンタ	LPT：		

＊1）デバイス名を省略した場合

・ドライブが内蔵または接続されているとき ････ ドライブA（A：）になる
・ドライブが内蔵や接続されていないとき ･････ カセットテープ（CAS：）になる

＊2）ファイル名に使用できる文字 英字、数字、カタカナ

ドライブ名

入出力機器にフロッピーディスクドライブを使用するときのデバイス名を、特に**「ドライブ名」**と呼びます。ドライブ名は、使用するドライブの接続のしかたにより異なります。

- **ドライブ１台が内蔵または接続されているとき** …… ドライブ名はA：になります。ただし、ＣＯＰＹ文などではB：として使うことができます。
- **ドライブ２台が内蔵されているとき** ……………… 片方がA：、他方がB：になります。
- **ドライブ１台が内蔵されているパソコンのスロットにドライブをもう１台接続したとき** ……… スロットに接続したドライブがA：、内蔵ドライブがC：になります。

拡張子について

拡張子は、ファイルの種類を区別するための分類記号です。文字数と使用可能な文字種以外、特に決められていません。使用するかどうか、どんな拡張子にするかは自由です。
たとえば下のように使い分けると、ファイルの数が多くなったときに便利です。

例)　ＢＡＳＩＣプログラム …… ＢＡＳ
　　　機械語プログラム ………… ＢＩＮ
　　　データファイル …………… ＤＡＴ

拡張子は省略できますが、拡張子をつけて保存したファイルを呼び出すときには省略できません。

ワイルドカード

ファイルを操作する命令（ＦＩＬＥＳやＣＯＰＹなど）では、ファイル名や拡張子に**「＊」**や**「？」**を使用することにより、複数のファイルを指定することができます。

＊（アスタリスク） …… ＊の直前までの文字が同じファイル名または拡張子をもつすべてのファイルを指定したことになります。
したがって「＊．＊」は、「そのディスク内のすべてのファイル」を指定したことになります。

？（クエスション） …… ？の部分以外の文字が同じファイル名または拡張子をもつすべてのファイルを指定したことになります。

このような機能を**「ワイルドカード」**と呼びます。

例)　ＦＩＬＥＳ "ＡＢＣ．＊" ……………… ＡＢＣ．ＢＡＳ
　　　　　　　　　　　　　　　　　　　ＡＢＣ．ＢＩＮ
　　　ＦＩＬＥＳ "？ＢＣ．ＢＡＳ" ………… ＡＢＣ．ＢＡＳ
　　　　　　　　　　　　　　　　　　　ＤＢＣ．ＢＡＳ
　　　ＣＯＰＹ "Ａ：＊．＊" ＴＯ "Ｂ："…… ディスク内の全ファイルをコピーする

保存と呼び出しの命令

BASICプログラムの保存・呼び出しの命令

・保存　SAVE, CSAVE（カセットテープ専用）
・呼び出し　LOAD, CLOAD（カセットテープ専用）, RUN, MERGE（アスキー型式のファイル専用）
・照合　CLOAD?（カセットテープ専用）

機械語プログラムやバイナリーデータ、VRAMデータの保存・呼び出しの命令

・保存　BSAVE, COPY（VRAMデータまたは配列変数領域とファイル間）
・呼び出し　BLOAD, COPY（ファイルとVRAMデータまたは配列変数領域間）

数値や文字列データの保存と呼び出しの命令

・保存　PRINT#, PRINT# USING
・呼び出し　INPUT#, INPUT$, LINE INPUT#

グラフィック画面やテキスト画面、プリンタへの文字の出力（出力専用）

PRINT#, PRINT# USING

シーケンシャルファイルとランダムファイル

データを保存するファイルには、シーケンシャルファイルとランダムアクセスファイルの2種類があり、どちらも一定の手順で操作することが必要です。

・シーケンシャルファイル

データを常に決まった順番で保存・呼び出しする方式のファイルです。ファイルの長さを自由に変えることができます（可変長のファイル型式）が、保存したデータを**一部分だけ変更したり削除することはできません**。フロッピーディスクでもカセットテープでも使用できます。

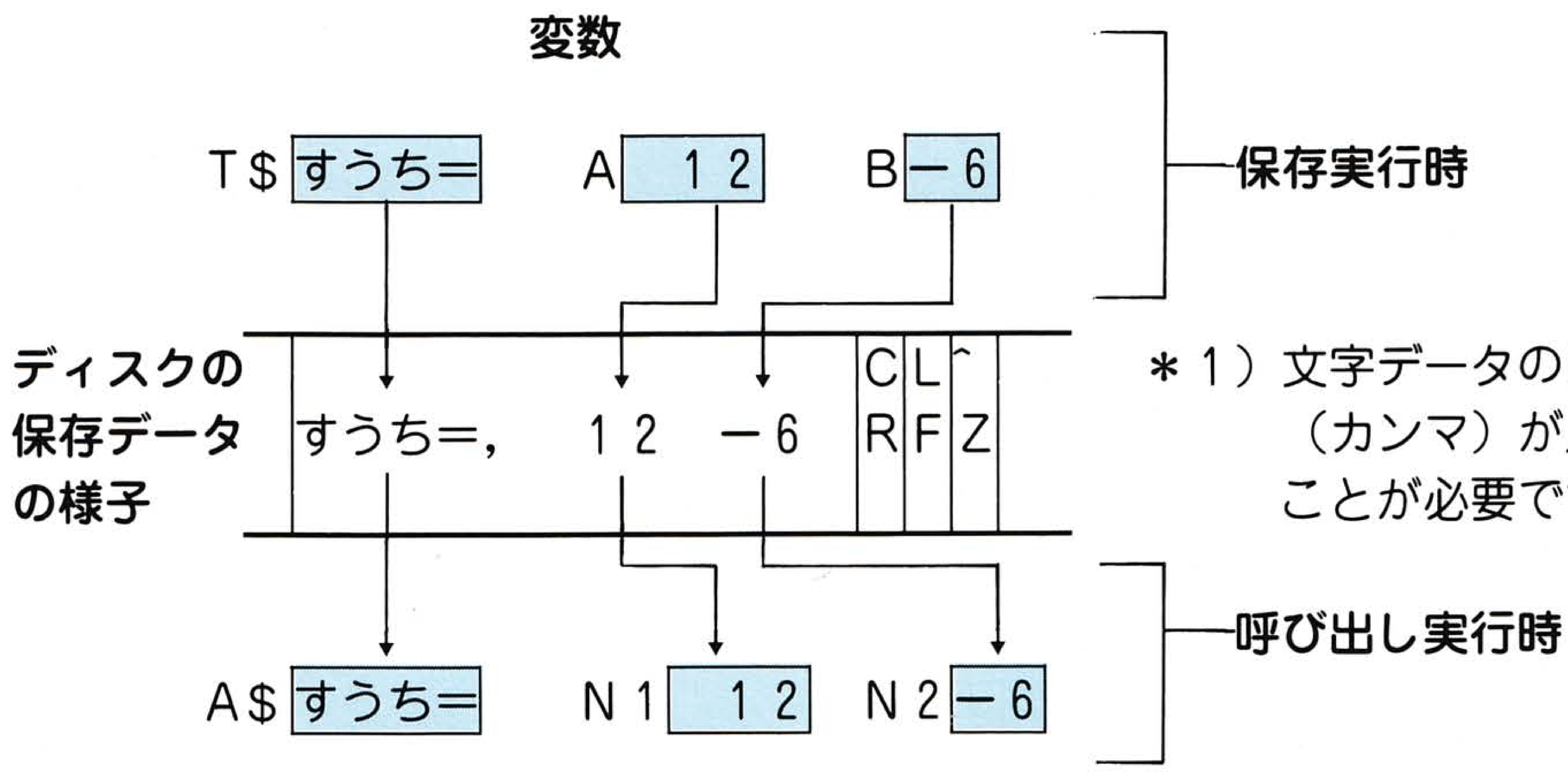
プログラム例）

保存

```
100 OPEN "A:FILE.DAT" FOR OUT
PUT AS #1
110 PRINT #1, T$;", ";A;B
120 CLOSE #1
```

呼び出し

```
100 OPEN "A:FILE.DAT" FOR INP
UT AS #1
110 INPUT #1, A$, N1, N2
120 CLOSE #1
```



＊1）文字データの区切りには、必ず「，」（カンマ）が入力されるようにすることが必要です。

＊2）複数のデータを保存・呼び出しする場合、ＯＰＥＮ，ＣＬＯＳＥを各データごとに実行する必要はありません。同じファイル番号を使う操作の最初と最後に一度ずつ実行するだけでかまいません。
ただし、操作中にディスクを入れ替えたり、ＥＮＤ命令以外でプログラムを終了する場合は事前にＣＬＯＳＥ命令を実行するようにしてください。これらの操作を、ファイルを開いたままで行うと、ファイルのデータが壊れることがあります。

・ランダムアクセスファイル

ファイルを一定の長さの部分（**「レコード」**）に区切って番号で管理することにより、保存や呼び出しを必要な部分から行えるようにしたファイルです。何らかのキー項目にしたがって保存位置を決定することにより、呼び出し時に必要なデータをすぐに取り出すことができます。
固定長のファイル型式になります。また、この型式のファイルは、**文字型のデータしかあつかうことができない**ので、数値型データを保存する場合は、あらかじめ文字型データに変換してから保存します。呼び出し時には、逆に文字型データから数値型データに変換してからデータを利用します。この方式のファイルは、**フロッピーディスクでだけ使用できます**。

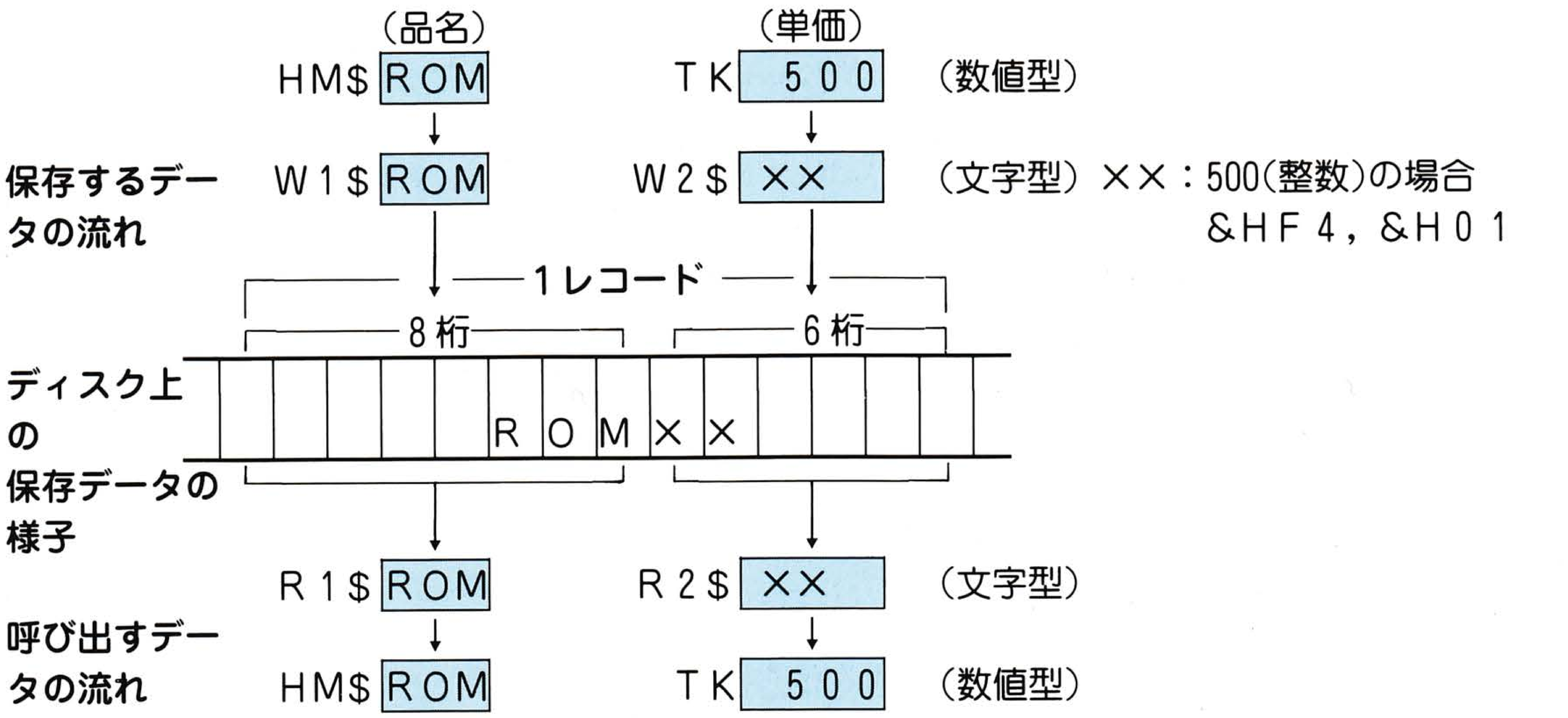
プログラム例）

保存

```
100 OPEN "A:FILE.DAT" AS #1
110 FIELD #1, 8 AS W1$, 6 AS W2$
120 RSET W1$=HM$
130 LSET W2$=MKI$(TK)            *1)
140 PUT #1, R                    *2)
150 CLOSE #1
```

呼び出し

```
100 OPEN "A:FILE.DAT" AS #1
110 FIELD #1, 8 AS R1$, 6 AS R2$
120 GET #1, R
130 HM$=R1$
140 TK=CVI(R2$)                  *3)
150 CLOSE #1
```



＊1）変数の型により、ＭＫＩ$，ＭＫＳ$，ＭＫＤ$関数のいずれかを使用します。
＊2）変数Ｒは、保存・呼び出しするデータの部分の番号（＝レコード番号：１以上の正の整数）として使っています。
＊3）変数の型により、ＣＶＩ，ＣＶＳ，ＣＶＤ関数のいずれかを使用します。

・複数のデータを保存・呼び出しする場合、ＯＰＥＮ，ＣＬＯＳＥを各データごとに実行する必要はありません。同じファイル番号を使う操作の最初と最後に一度ずつ実行するだけでかまいません。
・複数のファイルを同時に開く場合（＃１と＃２を同時に使用するときなど）には、事前にＭＡＸＦＩＬＥＳ文で同時に使用するファイル数の宣言が必要です。（**例** MAXFILES=2）ＭＡＸＦＩＬＥＳ文は、なるべくプログラムの先頭で実行させてください。プログラムの途中で実行させると、変数の値が失われる場合があります。
・ＦＩＬＥＤ文で割り当てることができるデータの長さの合計は、256バイトまたはＯＰＥＮ文で指定したレコード長（256バイト以下）までです。

RAMディスク

ＲＡＭディスクは、メインメモリの０～&Ｈ７ＦＦＦ番地をプログラムやデータの保存領域として一種の記憶装置のように使用する機能です。保存や呼び出しの操作のしかたはフロッピーディスクとほぼ同じです。（デバイス名は **MEM：** です⇨ 162ページ）
パソコンの電源スイッチを切ると、ＲＡＭディスクに保存したプログラムやデータは消えてしまいますが、保存や呼び出しの速度が非常に速いので、うまく使うと便利です。

ＲＡＭディスク用の拡張命令

・**CALL　MEMINI　（ＲＡＭディスクの上限の番地）**　……　ＲＡＭディスクの初期化
例）　CALL　MEMINI　&H4000
メモリ内にＲＡＭディスク用の領域を設定し、領域内のすべての記憶内容を消去する命令です。
ＲＡＭディスクを使用する場合は、電源スイッチを入れたあと必ず一度ＭＥＭＩＮＩ命令を実行させてください。
ＭＥＭＩＮＩ命令を実行させると、「×××××　ｂｙｔｅｓ　ｆｒｅｅ」というメッセージが表示されます。「×××××」の部分は、ＲＡＭディスクの上限の番地の指定に応じた数字で、ＲＡＭディスクの容量をあらわします。
ＲＡＭディスクの上限の番地は省略できます。その場合、ＲＡＭディスクの容量は最大の32,000バイトに設定されます。

・**CALL　MFILES**　……………………………　ファイル一覧の表示
・**CALL　MKILL　（"ファイルスペック"）**　………　ファイルの消去
・**CALL　MNAME　（"旧ファイルスペック"　AS　"新ファイルスペック"）**
ファイルスペックの変更

サブルーチンと割り込み

多くのプログラムは、いくつかの決まった手順の作業（処理）の組み合わせになります。プログラムの中である処理を繰り返して行う場合は、その処理を一か所に置いて必要が生じるたびにその部分にジャンプし、処理が終わったら元の部分に戻るようにすると、簡潔で理解しやすいプログラムが作れます。このようにした処理のことを**「サブルーチン」**と呼びます。
また、プログラムを実行中に特別なできごとが起きたとき、実行中の処理をただちに中断して別の処理を行い、その処理が終わったら元の処理の実行を再開することが必要な場合があります。
これを**「割り込み」**と呼びます。

サブルーチン

サブルーチンを使用するための命令として、次のものがあります。

GOSUB　行番号 ········ ＧＯＳＵＢ（ゴー・サブ）は、**行番号**でしめす行にジャンプし、その行から続く命令をＲＥＴＵＲＮ文にあうまで実行します。サブルーチンへジャンプするのに用います。
行番号は省略できません。存在しない行番号は指定できません。

RETURN　行番号 ······ ＲＥＴＵＲＮ（リターン）は、必ずＧＯＳＵＢ文と組み合わせて用いる命令です。サブルーチンの実行を終了するときに使用します。**行番号**は省略できます。省略したときはＧＯＳＵＢ文の次の文に戻って実行します。行番号を指定したときはその行番号にジャンプします。

ON　変数または計算式　GOSUB　行番号，行番号，行番号‥‥
ＯＮ　ＧＯＳＵＢ（オン～ゴー・サブ）は、変数の値によって異なるサブルーチンにジャンプする命令です。行番号の並びは左側から順番に１から始まる数に対応します。つまり、変数の値が３なら３番目の行番号にジャンプします。値が０または行番号の個数より大きいときは、ジャンプせずに次の行に進みます。値がマイナスのときはエラーになります。サブルーチンの実行を終了するときはＲＥＴＵＲＮ文を使用します。

・実行されるＧＯＳＵＢ文とＲＥＴＵＲＮ文の数が一致するように注意してください。対応するＧＯＳＵＢ文なしにＲＥＴＵＲＮ文を実行させるとエラーになります。
・サブルーチンから他のサブルーチンにジャンプすることもできます。

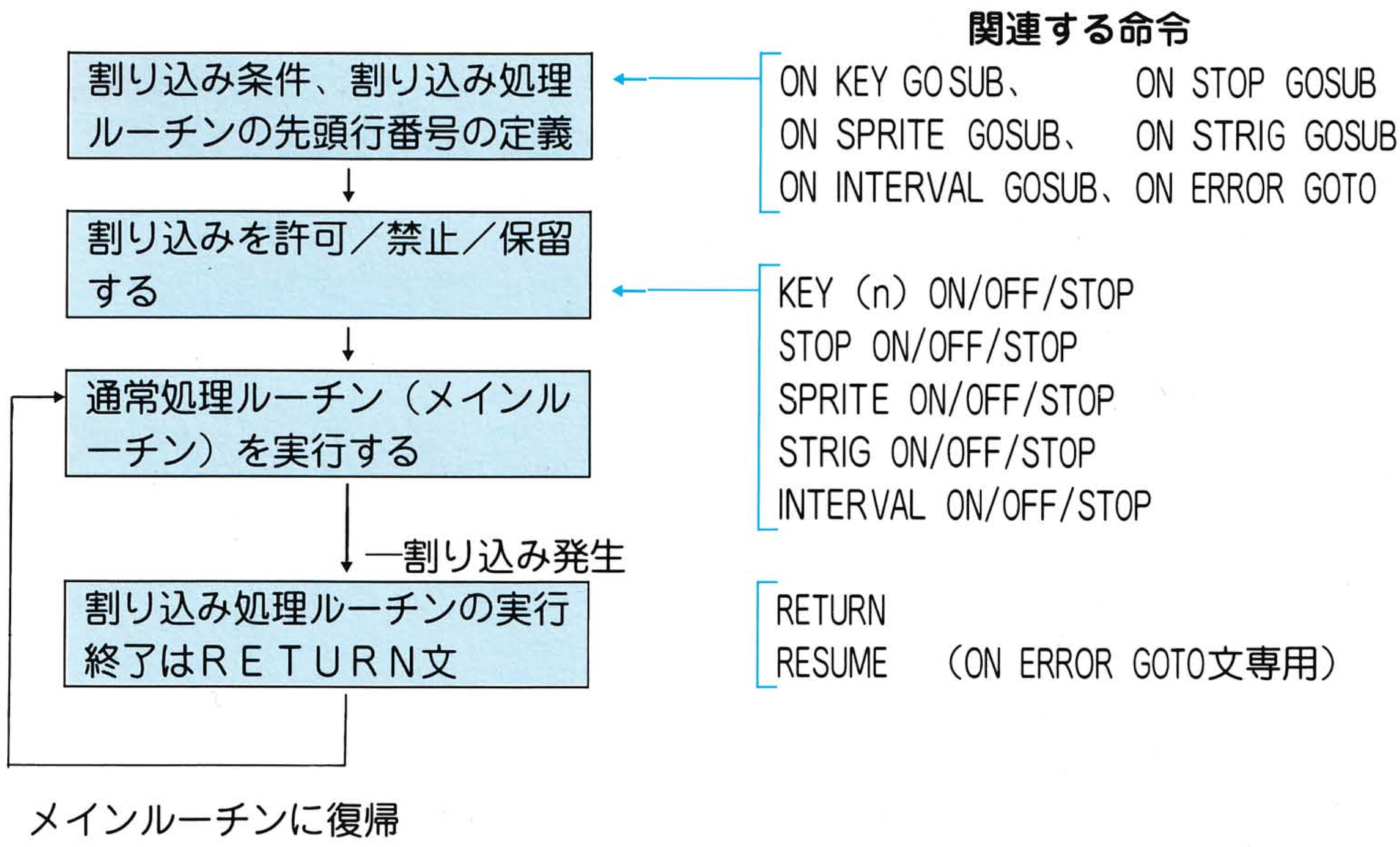
割り込み

通常、あることが起きたときに特定の処理を実行するときは、そのできごとが起きそうな部分でＩＮＫＥＹ$関数などを使って実際に起きているかどうかを調べます。しかし、これでは他の処理を実行中にそのできごとが起きたときには対処できません。そこでＢＡＳＩＣでは、一定のできごとが起きたとき、すぐに実行中の処理を中断して指定されたサブルーチンにジャンプするように設定する命令がいくつか用意されています。このような処理を**「割り込み」**と呼びます。

たとえばファンクションキーが押された時に、対応するサブルーチンにジャンプするようにあらかじめ設定しておくと、いつファンクションキーが押されてもその時点で割り込みがかかり、現在実行中の処理を中断して指定されたサブルーチン（割り込み処理ルーチン）にジャンプして新しい処理（たとえばミサイルの発射など）を実行したあと、中断していた処理を再開します。

ＭＳＸのＢＡＳＩＣで使用できる割り込みには、**ファンクションキー割り込み**、**ストップキー割り込み**、**スプライト衝突割り込み**、ジョイスティックなどの**トリガー割り込み**、**インターバルタイマー割り込み**があります。

割り込み使用時の処理の流れを下の図でしめします。



資料編

キャラクタコード表

上位４ビット（16進数）→

下位４ビット（16進数）↓

	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0	@	P	`	p	♠			ー	タ	ミ	た	み
1	!	1	A	Q	a	q	♥	あ	。	ア	チ	ム	ち	む
2	"	2	B	R	b	r	♣	い	「	イ	ツ	メ	つ	め
3	#	3	C	S	c	s	♦	う	」	ウ	テ	モ	て	も
4	$	4	D	T	d	t	○	え	、	エ	ト	ヤ	と	や
5	%	5	E	U	e	u	●	お	・	オ	ナ	ユ	な	ゆ
6	&	6	F	V	f	v	を	か	ヲ	カ	ニ	ヨ	に	よ
7	'	7	G	W	g	w	ぁ	き	ｧ	キ	ヌ	ラ	ぬ	ら
8	(	8	H	X	h	x	ぃ	く	ｨ	ク	ネ	リ	ね	り
9	)	9	I	Y	i	y	ぅ	け	ｩ	ケ	ノ	ル	の	る
A	*	:	J	Z	j	z	ぇ	こ	ｪ	コ	ハ	レ	は	れ
B	+	;	K	[	k	{	ぉ	さ	ｫ	サ	ヒ	ロ	ひ	ろ
C	,	<	L	¥	l	¦	ゃ	し	ｬ	シ	フ	ワ	ふ	わ
D	−	=	M	]	m	}	ゅ	す	ｭ	ス	ヘ	ン	へ	ん
E	.	>	N	^	n	~	ょ	せ	ｮ	セ	ホ	゛	ほ	
F	/	?	O	_	o		っ	そ	ｯ	ソ	マ	゜	ま	■

（&HFF ■ ←カーソル）

グラフィックキャラクタ

	4	5
0		π
1	月	┴
2	火	┬
3	水	┤
4	木	├
5	金	┼
6	土	│
7	日	─
8	年	┌
9	円	┐
A	時	└
B	分	┘
C	秒	╳
D	百	大
E	千	中
F	万	小

〈参考〉
特殊キーのキーコード

キー		コード
BS		&H8
TAB		&H9
HOME		&HB
RETURN		&HD
INS		&H12
SELECT		&H18
ESC		&H1B
カーソルキー	→	&H1C
	←	&H1D
	↑	&H1E
	↓	&H1F
DEL		&H7F

例）「A」のキャラクタコードは、上位４ビットが「４」、下位４ビットが「１」なので、
&Ｈ４１（10進数では、16×４＋１＝65）
「¥」のキャラクタコードは、上位４ビットが「５」、下位４ビットが「C」なので、
&Ｈ５Ｃ（10進数では、16×５＋12＝92：16進数のＡ～Ｆは10進数の10～15に対応）

グラフィックキャラクタの入出力

グラフィックキャラクタは、キャラクタコードの前にグラフィックヘッダ（&Ｈ０１）を付けて入出力されます。
たとえば、「時」のキャラクタコードは&Ｈ４Ａですから、
キーボードからの入力　　：&Ｈ０１、&Ｈ４Ａの２バイトが入力されます。
テレビ、プリンタへの出力：&Ｈ０１、&Ｈ４Ａの２バイトが出力されます。

例）テキスト画面へ「時」を表示　　PRINT　CHR$（1）;CHR$（&H4A）

BASICリファレンス表

以下のページでは、漢字ＢＡＳＩＣの命令や関数などを、用途別に分類して書式（入力の形式）、機能、文例（実際の入力例）を表にまとめています。

リファレンス表の見かた

必要な機器のマーク

特定の周辺機器を接続または内蔵していないと使用できない命令や関数については、以下のように必要な機器のマークを表示しています。

（フロッピーディスクのマーク）：フロッピーディスク	（カセットテープのマーク）：カセットテープ	（プリンタのマーク）：プリンタ	漢：漢字プリンタ

書　式

命令を入力するときの書きかたを示します。実際の入力時には、次のような決まりにしたがってください。

・**英字の大文字**で示された項目は、そのとおりに入力します。大文字でも小文字でもかまいません。

・**カギカッコ＜　＞**で示された項目は、ユーザーが指定する数値または文字列をあらわします。

・**角カッコ［　］**で示された項目は、省略することができます。省略したときは、初期値（ＢＡＳＩＣが設定する値）または最後に設定した値が適用されます。

・**棒線｜　｜**で囲まれた項目は、｜　｜内の項目から１つを選んで入力します。

・下のように**「，」（カンマ）**で区切られている項目を省略するときは、省略する項目の位置によって省略のしかたが変わります。

ＣＯＬＯＲ　［＜前景色＞］［，＜背景色＞］［，＜周辺色＞］

以後の項目をすべて省略するときは、「，」を含めて省略します。
後ろの項目を省略しないで入力するときは、それ以前の「，」を省略しないで入力します。

例）　ＣＯＬＯＲ　，，７

・**「;」（セミコロン）**で区切られている項目を省略するときは、「;」も含めて省略します。

・**「・・・」**は同じ項目の連続の省略をあらわしています。１行の許す長さ（ 255文字）の範囲内で任意の回数だけ繰り返すことができます。

・上記のカギカッコ、角カッコ、棒線、連続記号は入力してはいけません。それ以外の記号は、すべてそのまま入力します。

機　能　命令の機能を簡単に示します。

文　例　実際の入力のしかたの見本として基本的な入力例を示します。

命　令

プログラム作成

書　式	機　能	文　例
AUTO　[〈先頭の行番号〉][,〈増分値〉]	行番号を自動的に発生する	AUTO　100, 20
DELETE　[〈始めの行番号〉] －[〈終わりの行番号〉]	プログラムの一部を行単位で削除する	DELETE　50－100
KEY　LIST	ファンクションキーの内容を表示する	KEY　LIST
LIST　[〈先頭の行番号〉] －[〈最終の行番号〉]	メモリ上にあるプログラムのリストを表示する	LIST　100－200
LLIST　[〈先頭の行番号〉] －[〈最終の行番号〉]	メモリ上にあるプログラムのリストをプリンタで印刷する	LLIST
NEW	メモリ上のプログラムを消す	NEW
REM　[〈注釈文〉]　(「′」で代用可能)	プログラムに注釈を入れる	REM　*　これはサンプルです　*
RENUM [〈新先頭行番号〉] [,〈旧先頭行番号〉][,〈増分値〉]	プログラムの行番号を付け直す	RENUM　100, 10, 5

定義および設定

書　式	機　能	文　例
CALL ANK	BASICを英数・かなモードに設定する	CALL ANK
CALL KANJI[〈漢字モード〉]	BASICを漢字モードに設定する(モードは0～3)	CALL KANJI1
CALL MEMINI [〈サイズ〉]	RAMディスクの使用を宣言	CALL MEMINI
CLEAR [〈文字列領域の大きさ〉 [,〈メモリの上限〉]]	変数の初期化およびメモリ領域の大きさを設定する	CLEAR　300, &HE000
DEF　FN〈名前〉[(〈引数〉)] [,〈引数〉…])]=〈関数の定義式〉	関数を定義する	DEF　FNA(X, Y)=X*2/Y
DEF INT / SNG / DBL / STR 〈文字〉[－〈文字〉] [,〈文字〉[－〈文字〉]]…	頭文字で指定した変数の変数型を定義する (INT: 整数型、SNG: 単精度型 DBL: 倍精度型、STR: 文字型)	DEFINT　A－Z
DEF USR [〈番号〉]=〈開始アドレス〉	ユーザー作成の機械語サブルーチンの開始アドレスを定義	DEF　USR1=&HF000
DIM〈配列変数名〉(〈添字の最大値〉 [,〈添字の最大値〉…])	配列の名前、型、大きさ、次元を定義する	DIM　A(10, 20)
ERASE〈配列変数名〉[,〈配列変数名〉…]	配列変数を消去し、割り当てたメモリを解放する	ERASE　A, B, C
KEY〈キー番号〉,〈文字列〉	ファンクションキーに文字列を定義する	KEY1, "FILES"+ CHR$(13)

データ入出力(ファイル関係を除く)

書　式	機　能	文　例
DATA〈定数〉[,〈定数〉…]	READ文で読みとる数値や文字列を与える	DATA　123, ABC, キミ

書　式	機　能	文　例
ＩＮＰＵＴ［″〈プロンプト文〉″;］〈変数名〉［,〈変数名〉…］	キーボードからデータを入力し、指定した変数に入れる	ＩＮＰＵＴ　″すうりょうは″;N
ＬＩＮＥ　ＩＮＰＵＴ［″〈プロンプト文〉″;］〈文字型変数名〉	キーボードから文字列を入力し、文字型変数に入れる	ＬＩＮＥ　ＩＮＰＵＴ　″ナマエ″;A$
ＬＰＲＩＮＴ［〈式〉［,〈式〉］…］	プリンタで印刷する	ＬＰＲＩＮＴ　Ａ，Ｂ$;C
ＭＩＤ$（〈文字型変数〉,〈式1〉,［〈式2〉］）＝〈文字列〉	文字型変数の〈式1〉番目の文字から〈式2〉個の文字を〈文字列〉で置き換える	ＭＩＤ$（Ａ$，２，３）＝″ＢＣＤ″
ＲＥＡＤ〈変数名〉［,〈変数名〉…］	ＤＡＴＡ文からデータを読み取り、変数に入れる	ＲＥＡＤ　Ａ，Ｂ$，Ｃ$
ＲＥＳＴＯＲＥ［〈行番号〉］	次にＲＥＡＤ文で読み取るデータを指定する	ＲＥＳＴＯＲＥ　１０００

プログラムの実行・流れのコントロール

書　式	機　能	文　例
ＣＯＮＴ	CTRL ＋ STOP またはＳＴＯＰ文により停止させたプログラムの実行を再開	ＣＯＮＴ
ＥＮＤ	プログラムの実行を終了する。ＯＰＥＮしたままのファイルがあれば、ＣＬＯＳＥする	ＥＮＤ
ＦＯＲ〈変数名〉＝〈開始値〉ＴＯ〈終了値〉［ＳＴＥＰ〈増分値〉］ ＮＥＸＴ［［〈変数名〉］［,〈変数名〉］…］	ＦＯＲ文からＮＥＸＴ文までの一連の命令を、指定した回数だけ繰り返し実行	ＦＯＲ Ａ＝０ ＴＯ １５ ＳＴＥＰ ２ ： ＮＥＸＴ Ａ
ＧＯＳＵＢ〈行番号〉	サブルーチンにジャンプ	ＧＯＳＵＢ　１００
ＧＯＴＯ〈行番号〉	指定された行番号にジャンプ	ＧＯＴＯ　１００
ＩＦ〈条件〉ＴＨＥＮ｛〈文〉｜〈行番号〉｝［ＥＬＳＥ｛〈文〉｜〈行番号〉｝］ ＩＦ〈条件〉ＧＯＴＯ〈行番号〉［ＥＬＳＥ｛〈文〉｜〈行番号〉｝］	条件判断を行う。〈条件〉が真(-1)ならＴＨＥＮ文またはＧＯＴＯ文を実行。〈条件〉が偽(０)ならＥＬＳＥ文を実行。ＥＬＳＥ文を省略したときは、次の行に進む。	ＩＦ Ａ$＝″Ｙ″ ＴＨＥＮ ２００ ＥＬＳＥ ３００ （Ａ$が″Ｙ″なら200行へ Ａ$が″Ｙ″以外なら300行へ）
ＯＮ〈式〉ＧＯＳＵＢ〈行番号〉［,〈行番号〉…］	〈式〉の値によって異なるサブルーチンにジャンプする	ＯＮ　Ａ　ＧＯＳＵＢ　１００，２００，３００
ＯＮ〈式〉ＧＯＴＯ〈行番号〉［,〈行番号〉…］	〈式〉の値によって異なる行番号にジャンプする	ＯＮ　Ａ　ＧＯＴＯ　１００，２００，３００
ＲＥＴＵＲＮ［〈行番号〉］	サブルーチンから復帰する	ＲＥＴＵＲＮ
ＲＵＮ［〈行番号〉］	メモリ内のプログラムを実行行番号指定時はその行から	ＲＵＮ
(ディスク)(カセット) ＲＵＮ″〈ファイルスペック〉″	デバイス名で指定された機器からファイル名で指定されたプログラムをロードして実行	(ディスク) ＲＵＮ ″Ａ：ＰＲＯＧ１．ＢＡＳ″ (カセット) ＲＵＮ ″ＣＡＳ：ＰＲＯＧ１″
ＳＴＯＰ	プログラムの実行を停止してダイレクトモードに戻る	ＳＴＯＰ
ＴＲＯＦＦ	ＴＲＯＮを解除する	ＴＲＯＦＦ
ＴＲＯＮ	以後、プログラムを実行するたびに、実行した行番号をテキスト画面に表示する	ＴＲＯＮ

表示関係

■はテキスト画面専用。(g)はグラフィック画面専用。(e)は漢字テキスト画面では使えない。

書　式	機　能	文　例
CIRCLE [STEP](x, y),〈半径〉 [,〈カラーコード〉] [,〈開始角度〉] [,〈終了角度〉] [,〈比率〉] (g)	円またはだ円、扇形を描く (x, y)は中心座標	CIRCLE (80, 80) 40, 2
CALL CLS	画面を消去する	CALL CLS
CALL PALETTE [(〈カラーコード〉,〈R〉,〈G〉,〈B〉)]	カラーコードの色を変える 〈R〉〈G〉〈B〉は0～7の数値 項目を全部省略したときは、色を初期状態に戻す	CALL PALETTE (7, 7, 0, 0) (カラーコード7を赤に)
CLS (e)	画面を消去する	CLS
COLOR [〈前景色〉] [,〈背景色〉] [,〈周辺色〉]	画面の各部の色を設定する (初期値は、15,4,7)	COLOR 15, 2, 10
COLOR =(〈カラーコード〉,〈R〉,〈G〉,〈B〉)] (e)	カラーコードの色を変える 〈R〉〈G〉〈B〉は0～7の数値	COLOR=(8, 0, 7, 0) (カラーコード8を緑色に)
COLOR [=NEW] (e)	カラーコードの色を初期状態に戻す	COLOR
COLOR SPRITE (〈スプライト面番号〉)=〈カラーコード〉 (g)	指定したスプライト面のスプライトパターンの色を変える	COLOR SPRITE(0)=10
COLOR SPRITE$ (〈スプライト面番号〉)=〈カラーコード列〉 (g)	指定したスプライト面のスプライトパターンの1行毎の色を変える	COLOR SPRITE$(0) =CHR$(10)+CHR$(3)… (スプライトパターンの1行目をカラーコード10に、2行目をカラーコード3に)
DRAW〈文字列〉 (g)	画面上の参照点を相対的に移動させながら線を描く U,D,L,R,E,F,G,H,M,B,N,A,C,S,X;などのマクロ命令と数字からなる文字列で指定する	DRAW "C2U9L25D9R25"
KEY ON KEY OFF (t)	ファンクションキーの内容を画面最下行に表示するかしないかを指定する	KEY ON (表示する) KEY OFF (表示しない)
LINE [STEP] [(x1, y1)] -(x2, y2) [,〈カラーコード〉] [, {B / BF}] [,〈論理演算子〉] (g)	直線または四角形を描く Bをつけると2点を対角線とする四角形を描く BFをつけると四角形の内側を指定された色で塗りつぶす	LINE(0, 0)-(100, 100) , 2, B
LOCATE [〈桁〉] [,〈行〉] [,〈カーソルスイッチ〉] (t)	テキスト画面のカーソルの位置を指定する	LOCATE 14, 10
PAINT [STEP](x, y) [,〈領域色〉] [,〈境界色〉] (g)	指定された〈境界色〉で囲まれた領域を、指定された〈領域色〉で塗りつぶす	PAINT (50, 50), 8, 2 (境界色で囲まれていない領域を指定すると、画面全体を塗りつぶします)
PRESET [STEP](x, y) [,〈カラーコード〉] [,〈論理演算子〉] (g)	指定された位置に描かれた点を消す	PRESET (100, 100)
PRINT [〈式または文字列〉] [{, / ;}〈式または文字列〉…] (t)	文字列や数値をテキスト画面に表示する	PRINT "ABC", 123;"4"

書　式	機　能	文　例
PRINT USING〈書式〉; 〈式または文字列〉 [｜,｜〈式または文字列〉…] ｜;｜ (t) 書式制御文字 "!"　:最初の1字を表示 "&<n個の空白>&" :n+2字分の文字を表示 "@"　:文字列全部を表示 "#"　:数値の表示桁数を指定 " "　:小数点の位置指定桁ぞろえに使用 "+"　:数値の前(後)に+,-をつけて表示 "−"　:負の数値の後に−をつけて表示 （書式の右端につける） "**"　:#で指定した桁数に満たない数値の前を*で埋める （書式の左端につける） "¥¥"　:数値の前に¥をつけて表示 （書式の左端につける） "**¥"　:数値の前に¥をつけて表示し、#で指定した桁数に満たないときは数値の前を*で埋める （書式の左端につける） ","　:小数点から左に3桁毎に「,」を入れる（整数部の#の並びに置く） "^^^^"　:数値を指数形式で表示 （書式の右端につける）	文字列や数値を指定した書式でテキスト画面に表示する 〈書式〉は左の書式制御文字で指定する 数値の桁数が〈書式〉で指定した桁数より大きいときは、数値の右側に"%"が表示される 数値の小数部の桁数が指定より大きいときは、四捨五入して表示される	10 A$="ABCDE" 20 PRINT USING"!";A$ 30 PRINT USING"&　&";A$ 40 PRINT USING"XXX@YYY";A$ PRINT USING "###.##";123.456,789 PRINT USING "+###";100,−200 PRINT USING "###−";100,−200 PRINT USING "**###";10,−20 PRINT USING "¥¥###";10,−20 PRINT USING "**¥###";10,−20 PRINT USING "###,.##";1234.56 PRINT USING "#.##^^^^";123.45
PSET [STEP] (x,y) [,〈カラーコード〉] [,〈論理演算子〉] (g)	指定された位置に点を描く	PSET (100,100),8
PUT KANJI [STEP] (x,y) ,〈漢字コード〉[,〈カラーコード〉] [,〈論理演算子〉] [,〈モード〉] (g)	グラフィック画面に漢字を表示する	PUT KANJI (50,100) ,&H3441,3
PUT SPRITE〈スプライト面番号〉 [STEP] [,(x,y)][,〈カラーコード〉] [,〈スプライトパターン番号〉] (g)	指定されたスプライト面にスプライトパターンを表示する 同じ面に先に表示されていたパターンは消える	PUT SPRITE 0 ,(100,100),10
SCREEN [/画面モード〉] [,〈スプライトサイズ〉] [,〈キー音〉] [,〈カセットボーレート〉] [,〈プリンタの種類〉]	〈画面モード〉 0～12 〈スプライトサイズ〉 0:8×8ドット 1:8×8を2倍に拡大 2:16×16ドット 3:16×16を2倍に拡大 〈キー音〉 0:なし　1:あり 〈カセットボーレート〉 0:1200　1:2400 〈プリンタの種類〉 0:MSX標準プリンタ 1:非標準プリンタ	SCREEN 5,1

書　式	機　能	文　例
SET PAGE [〈表示ページ〉] [,〈描画ページ〉] (g)	スクリーン5～12の画面モードで、表示するVRAMのページと描画するVRAMのページを設定する	SET PAGE 0, 1
SET SCROLL [〈x〉] [,〈y〉] [,〈目隠し〉] [,〈次ページ〉]	表示画面を水平及び垂直にスクロールさせる	SET SCROLL 128, 100, 0, 0
WIDTH〈桁数〉 (t)	テキスト画面で1行に表示できる文字数を設定する	WIDTH 32

ファイル入出力

書　式	機　能	文　例
BLOAD "〈ファイルスペック〉" [,〈R〉] [,〈オフセット〉]	機械語プログラムをロードする Rがついているときは、ロード終了後実行する オフセットがついているときは、セーブ時の開始アドレスにオフセット分を加えたアドレスにロードする	BLOAD "PROG2. BIN", R BLOAD "CAS : PROG2", R
BSAVE "〈ファイルスペック〉" ,〈開始アドレス〉,〈終了アドレス〉 [, {〈実行開始アドレス〉 / S}]	〈開始アドレス〉から〈終了アドレス〉までのメモリの内容を、機械語プログラムとしてセーブする 実行開始アドレスを省略すると、〈開始アドレス〉から実行する Sをつけるとと VRAMの内容をディスクにセーブする	BSAVE "PROG3. BIN" , &HA100, &HA2FF BSAVE "CAS : PROG3" , &HA100, &HA2FF
CLOAD ["〈ファイル名〉"]	BASICのプログラムをカセットテープからロードする	CLOAD "PROG4"
CLOAD? ["〈ファイル名〉"]	カセットテープにセーブしたBASICのプログラムとメモリのプログラムを照合する	CLOAD? "PROG4"
CLOSE [[#]〈ファイル番号〉 [,[#]〈ファイル番号〉…]]	ファイルを閉じる	CLOSE #1, 2
CSAVE "〈ファイル名〉" [,〈ボーレート〉]	BASICのプログラムをカセットテープにセーブする	CSAVE "PROG4"
FIELD [#]〈ファイル番号〉 ,〈文字長〉 AS〈文字型変数〉 [,〈文字長〉 AS〈文字型変数〉…]	ランダムアクセスファイルの1レコードを分割する（合計 256文字またはレコード長以内）	FIELD #1, 16 AS A$, 4 AS B$, 24 AS C$
GET [#]〈ファイル番号〉 [,〈レコード番号〉]	ランダムアクセスファイルから1レコード入力する	GET #1, 2
INPUT #〈ファイル番号〉,〈変数名〉 [,〈変数名〉…]	シーケンシャルファイルからデータを順に読み取り、変数に入れる	INPUT #1, A, B$, C
LINE INPUT #〈ファイル番号〉 ,〈文字型変数名〉	1行単位のデータをシーケンシャルファイルから読み取り、文字型変数に入れる	LINE INPUT #1, A$

書　式	機　能	文　例
LOAD ″〈ファイルスペック〉″ [,R]	BASICのプログラムをロードする Rをつけるとロード後実行	LOAD ″CAS:PROG4″, R
LSET 〈文字型変数名〉=〈文字列〉	文字列をFIELD文で指定した変数に左づめで入れる	LSET A$=″タンカ″ LSET B$=D$
MAXFILES=〈ファイル数〉	同時にOPENする最大のファイル数を宣言する	MAXFILES=2
MERGE ″〈ファイルスペック〉″	アスキーセーブしたプログラムをロードしてメモリ内のプログラムと混合する	MERGE ″PROG5. ASC″
OPEN ″〈ファイルスペック〉″ [FOR 〈モード〉] AS #〈ファイル番号〉 [LEN=〈レコード長〉]	ファイルを開く 〈モード〉 ・シーケンシャルファイル OUTPUT…出力用 INPUT……入力用 APPEND…追加用 なし……テキスト画面 グラフィック画面 またはプリンタ ・ランダムファイル…なし	OPEN ″CAS:DATA1″ FOR INPUT AS #1 OPEN ″FILE1. DAT″ AS #1 グラフィック画面 OPEN ″GRP:″ AS #1
PRINT #〈ファイル番号〉 [{, ;} 〈式または文字列〉 [{, ;} 〈式または文字列〉]…]	シーケンシャルファイルにデータを出力する	PRINT #1, A, B, C
PRINT #〈ファイル番号〉 USING 〈書式〉;〈式または文字列〉 [[{, ;} 〈式または文字列〉]…]	指定された〈書式〉でシーケンシャルファイルにデータを出力する	PRINT #1 USING ″###. ##″;A$, B;C$
PUT [#]〈ファイル番号〉 [,〈レコード番号〉]	ランダムアクセスファイルに1レコード出力する	PUT #1, 1
RSET 〈文字型変数名〉=〈文字列〉	文字列をFIELD文で指定した変数に右詰めで入れる	RSET A$=″すうりょう″ RSET B$=D$
SAVE ″〈ファイルスペック〉″ [,A]	BASICのプログラムをセーブする デバイス名がドライブ名または省略されたときは、通常の形式でディスクにセーブ デバイス名がCAS:またはMEM:のときは、アスキーセーブ 末尾にAを入力されたときは、常にアスキーセーブ	SAVE ″CAS:PROG7″ SAVE ″PROG7. ASC″ SAVE ″PROG7. ASC″, A

〈ファイルスペック〉の形式

″[〈デバイス名〉:] [〈ファイル名〉] [. 〈拡張子〉]″

・**〈デバイス名〉**:は次のとおり。

CAS:……カセットテープ
CRT:……テキスト画面
GRP:……グラフィック画面
LPT:……プリンタ
MEM:……RAMディスク
A:〜H:…フロッピーディスクのドライブ名
省略………A:を指定したときと同じ

・**〈ファイル名〉**は、デバイスがフロッピーディスクまたはRAMディスクのときは8文字以内、カセットテープのときは6文字以内、それ以外は省略する。

・**〈拡張子〉**はデバイスがフロッピーディスクまたはRAMディスクのときに3文字以内でつける。それ以外は省略する。

ファイル管理

書　式	機　能	文　例
CALL FORMAT	ディスクをフォーマットする	CALL FORMAT
CALL MEMINI [(〈RAMディスクの上限〉)]	メモリ内にRAMディスク領域を設定し、初期化する	CALL MEMINI (3000)
CALL MFILES	RAMディスク内のファイルスペックを表示する	CALL MFILES
CALL MKILL ("〈ファイルスペック〉")	RAMディスク内のファイルを消去する	CALL MKILL ("PROG8.BAS")
CALL MNAME ("〈旧ファイルスペック〉" AS "〈新ファイルスペック〉")	RAMディスク内のファイルスペックを変更する	CALL MNAME ("PROG9" AS "PROG9.BAS")
COPY "〈ファイルスペック1〉" TO "〈ファイルスペック2〉"	ディスク内のファイルを同じディスクまたは他のディスクにコピーする	COPY "PROG1.BAS" TO "B:"
FILES ["〈ファイルスペック〉"]	ディスク内のファイルスペックを表示する	FILES FILES "*.BAS"
KILL "〈ファイルスペック〉"	ディスク内のファイルを消去する	KILL "PROG10.BAS"
LFILES ["〈ファイルスペック〉"]	ディスク内のファイルスペックをプリンタで印刷する	LFILES
NAME "〈旧ファイルスペック〉" AS "〈新ファイルスペック〉"	ディスク内のファイルスペックを変更する	NAME "PROG1.BAS" AS "GAME1.BAS"

割り込み

書　式	機　能	文　例
INTERVAL | ON | OFF | STOP |	タイマー割り込みを制御 ON　：許可する OFF　：禁止する STOP：保留する	INTERVAL ON INTERVAL OFF INTERVAL STOP
KEY (〈キー番号〉) | ON | OFF | STOP |	ファンクションキー割り込みを制御 ON　：許可する OFF　：禁止する STOP：保留する	KEY (1) ON KEY (2) OFF KEY (3) STOP
ON INTERVAL=〈時間〉 GOSUB 〈行番号〉	タイマー割り込みの条件を設定し、割り込み処理ルーチンの開始行を定義する	ON INTERVAL=60 GOSUB 100
ON KEY GOSUB [〈行番号〉] [,〈行番号〉…]	ファンクションキー割り込み処理ルーチンの開始行を定義	ON KEY GOSUB 100, 200, 300, 400, 500
ON SPRITE GOSUB 〈行番号〉	スプライトが重なったときに実行する割り込み処理ルーチンの開始行を定義する	ON SPRITE GOSUB 100
ON STOP GOSUB 〈行番号〉	CTRL + STOP 割り込み (CTRL を押しながら STOP を押す) 処理ルーチンの開始行を定義	ON STOP GOSUB 100

書　　式	機　能	文　例
ON STRIG GOSUB [〈行番号〉] [,〈行番号〉…]	SPACE やジョイスティックのトリガーボタンによる割り込みの処理ルーチンの開始行を定義する	ON STRIG GOSUB 100,200
SPRITE | ON | | OFF | | STOP |	スプライトが重なったときの割り込みを制御 ON　:許可する OFF　:禁止する STOP:保留する	SPRITE ON SPRITE OFF SPRITE STOP
STOP | ON | | OFF | | STOP |	CTRL ＋ STOP 割り込みの制御 ON　:許可する OFF　:禁止する STOP:保留する	STOP ON STOP OFF STOP STOP
STRIG (〈ジョイスティックの番号〉) | ON | | OFF | | STOP |	トリガーボタンによる割り込みを制御 ON　:許可する OFF　:禁止する STOP:保留する	STRIG ON STRIG OFF STRIG STOP

I／Oポート、メモリ関係

書　　式	機　能	文　例
OUT 〈ポート番号〉,〈式〉	I／Oポートに1バイトのデータを出力する	OUT &H22,&H80
POKE 〈アドレス〉,〈データ〉	メモリ内の指定されたアドレスにデータを書き込む	POKE &HF000,&HFF
VPOKE 〈アドレス〉,〈データ〉	VRAM内の指定されたアドレスにデータを書き込む	VPOKE &H8000,&HFF
WAIT 〈ポート番号〉,〈式1〉 [,〈式2〉]	I／Oポートからある値のデータが入力するまでプログラムの実行を中断する	WAIT 1,&H22,&H22

接続機器

書　　式	機　能	文　例
LPRINT [〈式〉]	文字列または数値をプリンタで印刷する	LPRINT ″123″;A,B$
LPRINT USING 〈書式〉; 〈式〉[| , | 〈式〉]… | ; |	文字列または数値を指定された書式でプリンタで印刷する (〈書式〉はPRINT USING参照)	LPRINT USING ″###″; A,B
LLIST [〈開始行番号〉] [−] [〈終了行番号〉]	プログラムのリストをプリンタで印刷する	LLIST 100−200
MOTOR [| ON |] | OFF |	データレコーダのモータをON／OFFする	MOTOR OFF

音に関する命令

書　　式	機　能	文　例
BEEP	ビープ音を鳴らす	BEEP
PLAY ″〈文字列1〉″ [,″〈文字列2〉″] [,″〈文字列3〉″]	音楽を演奏する（〈文字列〉はMMLと数字で構成）	PLAY ″O4CDEFGABO5C″ PLAY A$, B$+C$
SET BEEP 〈音色の種類〉 [,〈音量〉]	ビープ音の種類と大きさを設定する	SET BEEP 1, 3
SOUND 〈レジスタ番号〉,〈データ〉	PSG（音源IC）のレジスタに値を直接書き込む	SOUND 7, 254:SOUND 8, 8:SOUND 0, 255

●ミュージック・マクロ・ランゲージ（MML）

MML	機能と設定範囲	初期値
A[n]〜G[n]	音符（ド〜シがC,D,E,F,G,A,Bに対応）nは音符の長さ	――――
Ln	省略時の音の長さを設定　1≦n≦64	n＝4
Mn	音色のエンベロープ周期を設定する　1≦n≦65535	――――
Nn	音の高さを細かく設定　0≦n≦96	――――
On	オクターブを設定　1≦n≦8	n＝4
R[n]	休符　1≦n≦64	n＝4
Sn	音色のエンベロープ形状を設定する　0≦n≦15	――――
Tn	テンポの設定　32≦n≦255	n＝120
Vn	音量の設定　0≦n≦15	n＝8
X〈文字型変数〉;	〈文字型変数〉に入っているMMLを実行	――――
＝〈数値型変数〉;	〈数値型変数〉の値を直前のMMLのnにする	――――
.	直前の音符または休符の長さを1.5倍に　（付点音符）	――――
#, ＋	直前の音符を半音上げる　（シャープ）	――――
−	直前の音符を半音下げる　（フラット）	――――

エラー処理

書　　式	機　能	文　例
ERROR 〈エラー番号〉	指定したエラー番号のエラーを発生させる	ERROR 3
ON ERROR GOTO 〈行番号〉	エラーが発生したら、指定した行番号にジャンプする	ON ERROR GOTO 1000
RESUME [0 / NEXT / 〈行番号〉]	エラー処理ルーチンからメインプログラムに戻る 0および省略はエラーが発生した文に。 NEXTはその次の文に。	RESUME 10

その他の命令

書　　式	機　能	文　例
CALL 〈拡張命令〉	拡張命令を呼び出す	CALL MUSIC _MUSIC
CALL SYSTEM	MSX－DOSに制御を移す	CALL SYSTEM _SYSTEM
GET DATE 〈文字型変数〉 GET TIME 〈文字型変数〉	日付、時刻を変数に入れる	GET DATE D$ GET TIME T$
SET ADJUST (x,y)	画面の表示位置を調整する (x,yはそれぞれ－7～8)	SET ADJUST (－3, 5)
SET PASSWORD "〈パスワード〉"	MSX2＋画面にパスワードを設定する 次に表示されたときから、パスワードを入力するまで次の画面に移らない パスワードを忘れたときは次のようにする ①内蔵ソフト切り替えスイッチがあるパソコンでは、あらかじめ「切」にする ② GRAPH と STOP を押しながらリセットボタンを押し、BASICの画面が表示されるまで押し続ける ③ STOP を押してキー入力を可能にする ④下の文を入力する SET TITLE ""	SET PASSWORD "AAA"
SET PROMPT "〈プロンプト〉"	ダイレクトモードの「Ok」を他の文字に変更する	SET PROMPT "Ready!"
SET TITLE "〈タイトル〉" [,〈背景色〉]	MSX2＋画面の背景色(1～4)とMSX2＋マークの下に表示される〈タイトル〉を指定	SET TITLE "Hello", 2
SET SCREEN	実行時のSCREEN命令の設定をSRAMに記録する 次からスタート時に同じ設定になる	SET SCREEN
SET DATE "〈日付〉" SET TIME "〈時刻〉"	日付や時刻を設定する	SET DATE "89/01/15" SET TIME "12:34:56"

関　数

以下、ｎは数値、数値型変数または計算式をあらわします。

数値型関数

書　　式	機　能	文　例
ＡＢＳ（ｎ）	絶対値を返す	Ａ＝ＡＢＳ（Ｂ）
ＡＴＮ（ｎ）	アークタンジェントを返す （ｎはπを単位とする角度）	Ａ＝ＡＴＮ（Ｂ／１８０＊３．１４１５）
ＣＡＬＬ　ＫＩＮＳＴＲ（〈数値型変数〉 [，ｎ]，〈文字列1〉，〈文字列2〉） **注：**漢字モードでのみ使用可能	〈文字列1〉のｎ文字目から後ろの〈文字列2〉の位置を、見つからなければ０を変数に入れる ｎの省略時は１文字目から	ＣＡＬＬ　ＫＩＮＳＴＲ （Ａ，１０，Ｂ＄，″愛″）
ＣＡＬＬ　ＫＬＥＮ （〈数値型変数〉，〈文字列〉[，〈機能〉]） **注：**漢字モードでのみ使用可能	〈機能〉が０（または省略）のときは、〈文字列〉の全体の長さを、１のときは半角文字の数を、２のときは全角文字の数を変数に入れる	ＣＡＬＬ　ＫＬＥＮ（Ａ，Ｂ＄，２）
ＣＡＬＬ　ＫＴＹＰＥ （〈数値型変数〉，〈文字列〉，ｎ） **注：**漢字モードでのみ使用可能	〈文字列〉中のｎ番目の文字のタイプ（半角なら０，全角なら１）を変数に入れる	ＣＡＬＬ　ＫＴＹＰＥ　（Ａ，Ｂ＄，１）
ＣＤＢＬ（ｎ）	倍精度実数に変換して返す	Ａ＝ＣＤＢＬ（Ｂ％）
ＣＩＮＴ（ｎ）	整数に変換して返す	Ａ％＝ＣＩＮＴ（Ｂ）
ＣＯＳ（ｎ）	コサインを返す （ｎはラジアン単位の角度）	Ａ＝ＣＯＳ（Ｂ／１８０＊３．１４１５）
ＣＳＮＧ（ｎ）	単精度実数に変換して返す	Ａ！＝ＣＳＮＧ（Ｂ）
ＥＲＬ	発生したエラーのエラー番号を返す	Ａ＝ＥＲＬ ＩＦ　Ａ＝６８　ＴＨＥＮ　１００
ＥＲＲ	エラーが発生した行の行番号を返す	Ａ＝ＥＲＬ ＩＦ　Ａ＝４０　ＴＨＥＮ　２００
ＥＸＰ（ｎ）	e^nを返す	Ａ＝ＥＸＰ（Ｂ）
ＦＩＸ（ｎ）	ｎの整数部分を返す	Ａ＝ＦＩＸ（Ｂ）
ＩＮＴ（ｎ）	ｎ未満の最大の整数を返す	Ａ＝ＩＮＴ（Ｂ）
ＬＯＧ（ｎ）	自然対数を返す	Ａ＝ＬＯＧ（Ｂ）
ＲＮＤ［（ｎ）］	０以上１未満の乱数を返す ｎが０（または省略）のとき直前に発生した乱数の値、 ｎが正のときはＲＵＮ文を実行するたびに同じ系列の乱数、 ｎが負のときはｎの値によって決まった値を発生。負のときは同時に乱数系列を初期化	Ａ＝ＲＮＤ（－Ｂ）
ＳＧＮ（ｎ）	ｎが正なら１、負ならー１、０なら０を返す	Ａ＝ＳＧＮ（Ｂ）
ＳＩＮ（ｎ）	サインを返す （ｎはラジアン単位の角度）	Ａ＝ＳＩＮ（Ｂ／１８０＊３．１４１５）
ＳＱＲ（ｎ）	平方根を返す	Ａ＝ＳＱＲ（Ｂ）
ＴＡＮ（ｎ）	タンジェントを返す （ｎはラジアン単位の角度）	Ａ＝ＴＡＮ（Ｂ）

文字型関数

書　　式	機　能	文　例
CALL AKCNV (〈文字型変数〉,〈文字列〉) **注：**漢字モードでのみ使用可能	〈文字列〉のすべての文字を全角文字に変換し、〈文字型変数〉に入れる	CALL　AKCNV (A$, B$)
CALL JIS (〈文字型変数〉,〈文字列〉) **注：**漢字モードでのみ使用可能	〈文字列〉の最初の２バイトをJISコードに変換して〈文字型変数〉に入れる	CALL　JIS (A$, B$)
CALL SJIS (〈文字型変数〉,〈文字列〉) **注：**漢字モードでのみ使用可能	〈文字列〉の最初の２バイトをシフトJISコードに変換して〈文字型変数〉に入れる	CALL　SJIS (A$, B$)
CALL　KACNV (〈文字型変数〉,〈文字列〉) **注：**漢字モードでのみ使用可能	〈文字列〉の半角文字に変換できる文字をすべて半角文字に変換し、〈文字型変数〉に入れる	CALL　KACNV (A$, B$)
CALL　KEXT (〈文字型変数〉,〈文字列〉,〈機能〉) **注：**漢字モードでのみ使用可能	〈機能〉が０なら〈文字列〉の半角文字を、１なら全角文字を〈文字型変数〉に入れる	CALL　KEXT (A$, B$, 1)
CALL　KMID (〈文字型変数〉〈文字列〉, n1 [, n2]) **注：**漢字モードでのみ使用可能	〈文字列〉の左からn1番目の文字からn2個の文字を〈文字型変数〉に入れる （n2省略時は終わりまで）	CALL　KMID (A$, B$, 10, 3)
CALL　KNJ (〈文字型変数〉,"〈漢字コード〉") **注：**漢字モードでのみ使用可能	「"」で囲んだ４桁の数字で指定される漢字コードに相当する全角文字を〈文字型変数〉に入れる	CALL　KNJ (A$, "3441") CALL　KNJ (A$, B$)
LEFT$ (〈文字列〉, n)	〈文字列〉の左からn個の文字を返す	A$=LEFT$ (B$, 3)
MID$ (〈文字列〉, n1 [, n2])	〈文字列〉の左からn1番目の文字からn2個の文字を返す （n2省略時は終わりまで）	A$=MID$ (B$, 5, 1)
RIGHT$ (〈文字列〉, n)	〈文字列〉の左からn個の文字を返す	A$=MID$ (B$, 3)
SPACE$ (n)	n個の空白を文字列として返す	A$=SPACE$ (12)
SPC (n)	n個の空白を出力する	PRINT　SPC (12)
STRING$ (n, |〈文字列〉 〈キャラクタコード〉|)	〈文字列〉の最初の文字または〈キャラクタコード〉(0～255) であらわされる文字をn個分出力する	STRING$ (10, "+")
TAB (n)	指定された桁まで空白を出力	TAB (10)

数値型⇆文字型変換に関する関数

書　式	機　能	文　例
ASC（〈文字列〉）	〈文字列〉の先頭の文字のキャラクタコードを返す	A＝ASC（B$）
BIN$（n）	nの2進表記を文字列として返す	A＝BIN$（15）
CALL KINSTR（〈数値型変数〉 [,n],〈文字列1〉,〈文字列2〉） 注：漢字モードでのみ使用可能	〈文字列1〉のn文字目から後ろの〈文字列2〉の位置を、見つからなければ0を変数に入れる nの省略時は1文字目から	CALL KINSTR （A，10，B$，"愛"）
CALL KLEN （〈数値型変数〉,〈文字列〉[,〈機能〉]） 注：漢字モードでのみ使用可能	〈機能〉が0（または省略）のときは、〈文字列〉の全体の長さを、1のときは半角文字の数を、2のときは全角文字の数を変数に入れる	CALL KLEN（A，B$，2）
CALL KTYPE （〈数値型変数〉,〈文字列〉,n） 注：漢字モードでのみ使用可能	〈文字列〉中のn番目の文字のタイプ（半角なら0，全角なら1）を変数に入れる	CALL KTYPE（A，B$，1）
CHR$（n）	nをキャラクタコードとしてもつ文字を返す	A$＝CHR$（B）
CVI CVS CVD	ランダムアクセスファイルから入力した文字型データを数値に変換して返す	A＝CVI（A$） B＝CVS（B$） C＝CVD（C$）
MKI$ MKS$ MKD$	ランダムアクセスファイルに保存するため、数値を文字型データに変換する	A$＝MKI$（A） B$＝MKS$（B） C$＝MKD$（C）
HEX$（n）	nの16進表記を文字列として返す	A$＝HEX$（15）
INSTR（[n,]〈文字列1〉,〈文字列2〉）	〈文字列1〉のn個目の文字から後ろで〈文字列2〉を探し、その位置を返す	INSTR（2，A$，B$）
LEN（〈文字列〉）	〈文字列〉の文字数を返す	A＝LEN（B$）
OCT$（n）	nの8進表記を文字列として返す	A$＝OCT$（15）
STR$（n）	数値をあらわす文字列を返す	A$＝STR$（20）
VAL（〈文字列〉）	文字列があらわす数値を返す	A＝VAL（"15"）

データの入力に関する関数

●スクリーンからの入力

書　式	機　能	文　例
CSRLIN　（システム変数）	カーソルのタテ座標を返す	A＝CSRLIN
POS（ダミーの数値）　（システム変数）	カーソルのヨコ座標を返す	A＝POS（0）
POINT（x，y）	（x，y）の位置の点のカラーコードを返す	A＝POINT（50，100）

●データファイルからの入力

書　式	機　能	文　例
EOF（〈ファイル番号〉）	ファイルの最後のデータがすでに入力されていたら−1、それ以外は0を返す	A＝EOF（1） IF　A＝−1　THEN…
INPUT$（n [,[#]〈ファイル番号〉]）	ファイルからn文字入力して返す	A$＝INPUT$（6，#1）
LOF（〈ファイル番号〉）	ファイルの長さをバイト単位で返す	A＝LOF（1）
LOC（〈ファイル番号〉）	ランダムアクセスファイルのときは、現在のレコード番号を返す シーケンシャルファイルのときは、OPENしてから入出力したレコード数（INPUTモードではレコード数＋1）を返す。	A＝LOC（1）

●プリンタからの入力

書　式	機　能	文　例
LPOS（〈ダミーの数値〉）	プリンタバッファ内のプリンタヘッドの位置を返す	A＝LPOS（0）

●メモリからの入力

書　式	機　能	文　例
FRE（｛〈ダミーの数値〉 〈ダミーの文字列〉｝）	引数が〈ダミーの数値〉のときは、メモリの未使用領域の大きさを返す 〈ダミーの文字列〉のときは、メモリの文字列領域の未使用部分の大きさを返す	A＝FRE（0） A＝FRE（""）
PEEK（〈アドレス〉）	メモリの指定されたアドレスの内容を返す	A＝PEEK（&HA000）
VARPTR（〈変数名〉）	指定された変数の値が格納されている先頭アドレスを返す	A＝VARPTR（B）
VARPTR（#〈ファイル番号〉）	指定されたファイルのファイルコントロールブロックの先頭アドレスを返す	A＝VARPTR（#1）
VPEEK（〈アドレス〉）	VRAMの指定されたアドレスの内容を返す	A%＝VPEEK（&H1000）

●キーボードからの入力

書　式	機　能	文　例
INKEY$	現在、押されているキーの文字を返す（何も押されていなければ""を返す）	A$＝INKEY$
INPUT$（n）	キーボードからn文字入力する	A$＝INPUT$（3）

●ディスクからの入力

書　　式	機　能	文　例
DSKF（《ドライブ番号》）	番号で指定されたドライブ内のディスクの残り容量をクラスタ単位で返す 0と1…ドライブA： 2………ドライブB： ・ ・ 8………ドライブH：	PRINT　DSKF（1）

●I／Oポートからの入力

書　　式	機　能	文　例
INP（《ポート番号》）	I／Oポートから入力しているデータを返す	A＝INP（15）

●機械語サブルーチンからの入力

書　　式	機　能	文　例
USR［n］（《引数》）	指定された番号（0～9）のユーザーサブルーチンを実行し、結果を返す	A＝USR　1（2）

●タッチパッド（マウス）、パドル、ジョイスティック（ジョイパッド）からの入力

書　　式	機　能	文　例
PAD（n）	タッチパッドなどの状態を返す（マウスの場合のみ説明） n＝12または16のとき 　マウスに対する入力要求 　（必ず－1を返す） n＝13または17のとき 　マウスのヨコ座標 n＝14または18のとき 　マウスのタテ座標 （トラックボールも同じ）	A＝PAD（12）
PDL（n）	パドルの状態を返す	A＝PDL（1）
STICK（n）	nで指定されたジョイスティックの方向を数値で返す （nが0ならカーソルキー） 0：中央 1：上　　5：下 2：右上　6：左下 3：右　　7：左 4：右下　8：左上	A＝STICK（0）
STRIG（n）	nで指定されたジョイスティックなどのトリガーボタンが押されていれば1を、押されていなければ0を返す （nが0なら SPACE キー）	A＝STRIG（0）

定数と変数

定数	文字型	0～255文字の文字列。「″」（ダブルクォート）で囲む。
	整数型	−32768～＋32767
	固定小数点型	小数点を含む14桁までの数値（実数）。14桁を超える数値の表示は、浮動小数点型になる。
	浮動小数点型	Ｅを使った指数で表示。仮数部の有効数字14桁（倍精度）または６桁（単精度）。 指数部の範囲　−64～＋62
	16進形式	＆Ｈを数字の前につける。
	８進形式	＆Ｏを数字の前につける。
	２進形式	＆Ｂを数字の前につける。

変数	変数名	先頭の２文字のみ有効。（ただし、型宣言文字や配列変数の添え字は変数名に含めない）
	型宣言文字	変数名の後ろにつける。 ％：整数型　＃：倍精度　！：単精度　＄：文字型
	配列変数	変数名の後ろに（　）で囲んで添え字をつける。添え字が10を超える場合は事前に宣言が必要。 添え字は０から始まる整数。各次元の添え字のあいだは「，」で区切る。

特別な変数(システム変数)

書　　式	機　能	文　　例
ＢＡＳＥ（ｎ）	画面出力に関するテーブルのアドレスを返す。ｎは０～44	Ａ＝ＢＡＳＥ（１０）
ＳＰＲＩＴＥ＄＝〈文字列〉［＋〈文字列〉…］	スプライトパターンを定義する ８×８ドットのとき‥８文字 16×16ドットのとき‥32文字	SPRITE$=CHR$(24)+CHR$(60)+CHR$(90)+CHR$(126)+CHR$(255)+CHR$(36)+CHR$(66)+CHR$(66) SPRITE$=″″
ＴＩＭＥ	内蔵タイマーの値を示す 数値を入れて書き換え可能	ＰＲＩＮＴ　ＴＩＭＥ ＴＩＭＥ＝０
ＶＤＰ（〈レジスタ番号〉）	ＶＤＰのレジスタを直接参照更新する	Ａ＝ＶＤＰ（８）

演算記号

算術演算記号	^	累乗（べき）	ＭＯＤ	整数剰余
	−	正負符号を反転	＋，−	加算、減算
	＊，／	乗算、除算	（上から順に優先。ただし式中に「かっこ」があるときは、内側のかっこの中から先に演算する）	
	¥	整数除算		
関係演算記号	＞，＜，＝	大小比較。関係が真のときは演算結果の値が−１、偽のときは値が０になる		
論理演算記号	ＮＯＴ	否定	ＸＯＲ	排他的論理和
	ＡＮＤ	論理積	ＥＱＶ	排他的論理和の否定
	ＯＲ	論理和	ＩＭＰ	含意

カラーコード(初期設定時)

コード	色
0	透　明
1	黒
2	緑
3	淡い緑

コード	色
4	濃い青
5	青
6	濃い赤
7	水　色

コード	色
8	赤
9	淡い赤
10	黄
11	淡い黄

コード	色
12	濃い緑
13	紫
14	灰
15	白

BASIC予約語表

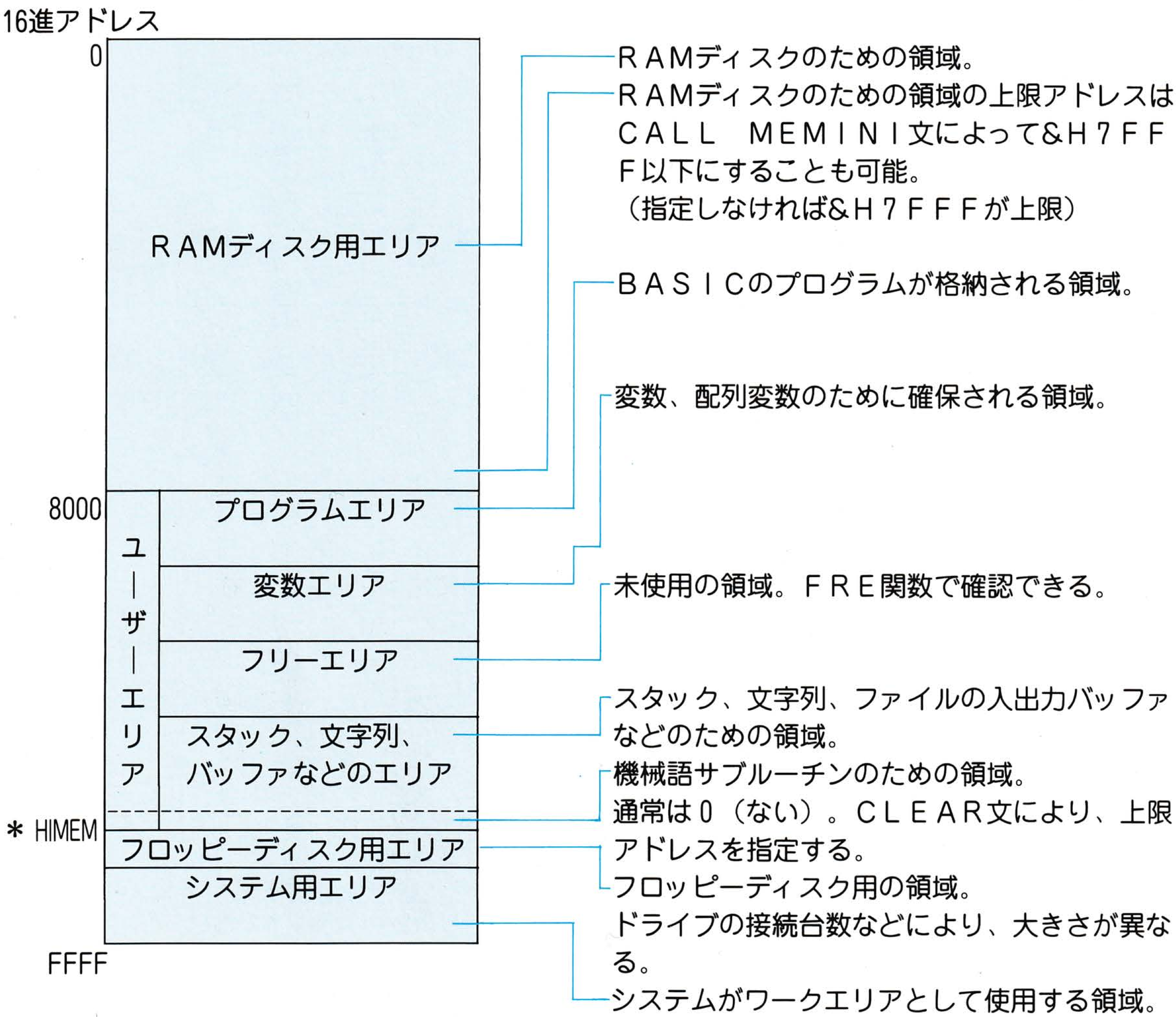
ABS
AND
ASC
ATN
ATTR$
AUTO
BASE
BEEP
BIN$
BLOAD
BSAVE
CALL
CDBL
CHR$
CINT
CIRCLE
CLEAR
CLOAD
CLOSE
CLS
CMD
COLOR
CONT
COPY
COS
CSAVE
CSNG
CSRLIN
CVD
CVI
CVS
DATA
DEF
DEFDBL
DEFINT
DEFSNG
DEFSTR
DELETE
DIM
DRAW
DSKF
DSKI$
DSKO$
ELSE
END
EOF
EQV
ERASE
ERL
ERR
ERROR
EXP
FIELD
FILES
FIX
FN
FOR
FPOS
FRE
GET
GO TO
GOSUB
GOTO
HEX$
IF
IMP
INKEY$
INP
INPUT
INSTR
INT
IPL
KEY
KILL
LEFT$
LEN
LET
LFILES
LINE
LIST
LLIST
LOAD
LOC
LOCATE
LOF
LOG
LPOS
LPRINT
LSET
MAX
MERGE
MID$
MKD$
MKI$
MKS$
MOD
MOTOR
NAME
NEW
NEXT
NOT
OCT$
OFF
ON
OPEN
OR
OUT
PAD
PAINT
PDL
PEEK
PLAY
POINT
POKE
POS
PRESET
PRINT
PSET
PUT
READ
REM
RENUM
RESTORE
RESUME
RETURN
RIGHT$
RND
RSET
RUN
SAVE
SCREEN
SET
SGN
SIN
SOUND
SPACE$
SPC(
SPRITE
SQR
STEP
STICK
STOP
STR$
STRIG
STRING$
SWAP
TAB(
TAN
THEN
TIME
TO
TROFF
TRON
USING
USR
VAL
VARPTR
VDP
VPEEK
VPOKE
WAIT
WIDTH
XOR

コントロールコード表

コード（10進）	コード（16進）	機　能	対応キー
0	00		
1	01	グラフィックキャラクタの入出力時のヘッダ	CTRL + A
2	02	カーソルを直前の語の先頭に移動	CTRL + B
3	03	入力待ち状態を終了する	CTRL + C
4	04		CTRL + D
5	05	カーソル以下を削除	CTRL + E
6	06	カーソルを次の語の先頭へ移動	CTRL + F
7	07	ＢＥＥＰ音を鳴らす（ＢＥＥＰ文と同じ）	CTRL + G
8	08	カーソルの１つ前の文字を削除する	CTRL + H, BS
9	09	次の水平タブ位置に移動	CTRL + I, TAB
10	0A	行を送る（ラインフィード）	CTRL + J
11	0B	カーソルをホームポジション（画面左上角）に戻す	CTRL + K
12	0C	画面を消去し、カーソルをホームポジションに戻す	CTRL + L
13	0D	カーソルを画面の左端に戻す（キャリッジリターン）	CTRL + M, ↵
14	0E	カーソルを現在の行の行末に移動	CTRL + N
15	0F		CTRL + O
16	10		CTRL + P
17	11		CTRL + Q
18	12	挿入モードのＯＮ／ＯＦＦスイッチ	CTRL + R, INS
19	13		CTRL + S
20	14		CTRL + T
21	15	現在カーソルがある１行を画面から削除	CTRL + U
22	16		CTRL + V
23	17		CTRL + W
24	18		CTRL + X, SELECT
25	19		CTRL + Y
26	1A		CTRL + Z
27	1B		CTRL + [, ESC
28	1C	カーソルを右へ１文字移動	CTRL + ¥, →
29	1D	カーソルを左へ１文字移動	CTRL +], ←
30	1E	カーソルを上へ１文字移動	CTRL + ^, ↑
31	1F	カーソルを下へ１文字移動	CTRL + _, ↓
127	7F	カーソルの位置の１文字を削除	DEL

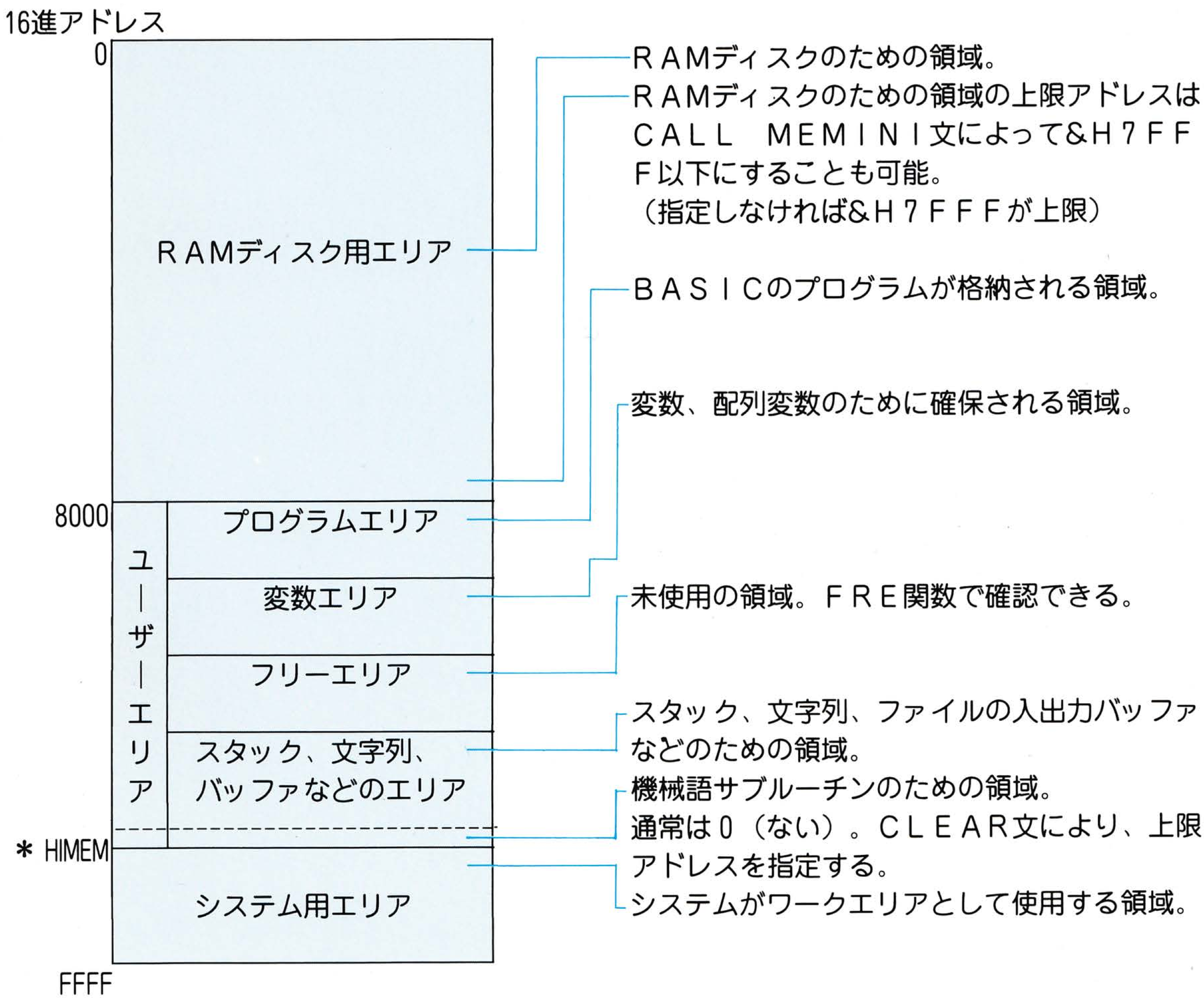
メモリマップ(BASIC使用時)

フロッピーディスクドライブを内蔵または接続している場合



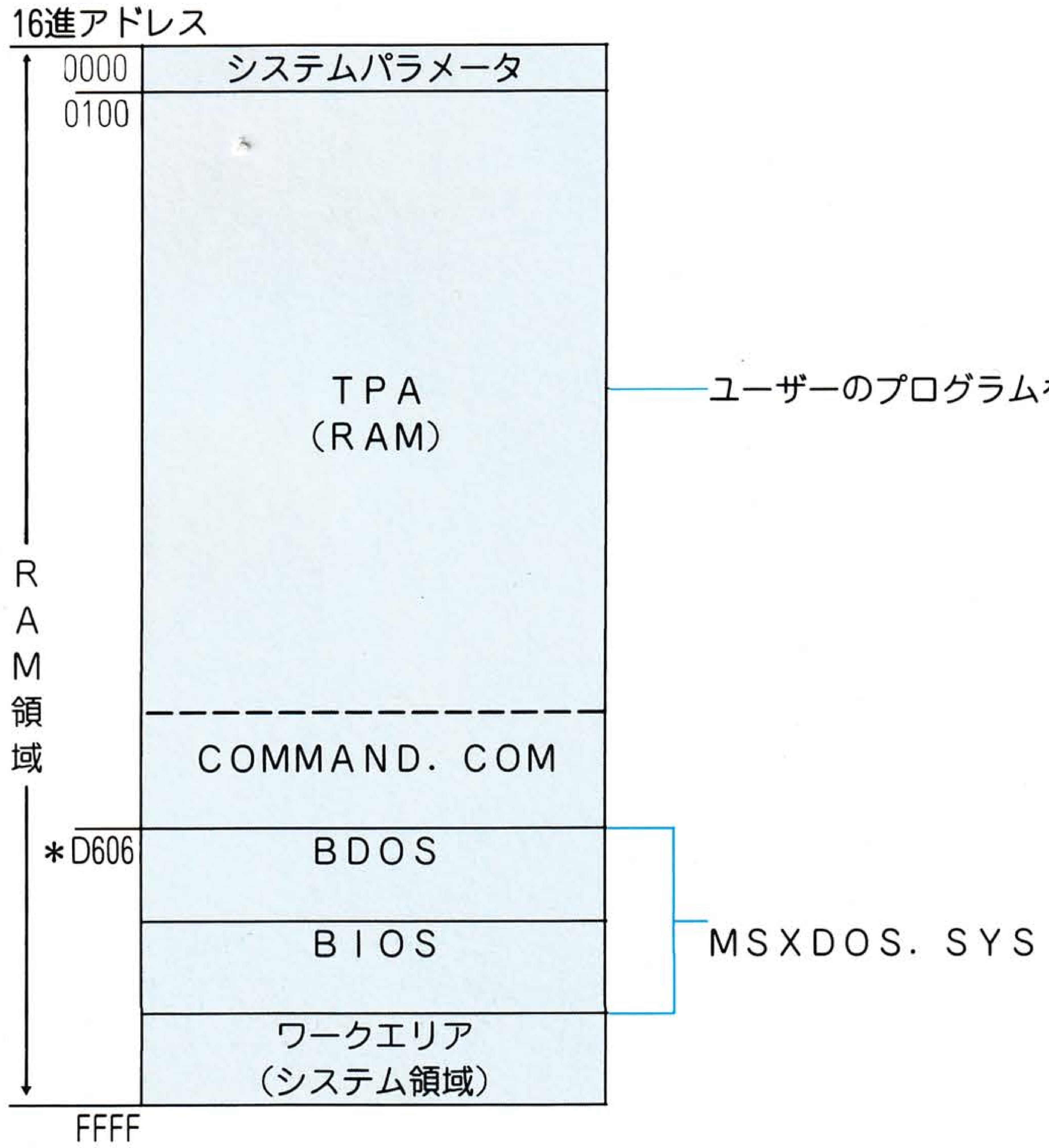
＊HIMEMのアドレスは、&HFC4Aと&HFC4B番地に格納されている。
HIMEM＝（&HFC4B）×&H100＋（&HFC4A）

フロッピーディスクドライブを内蔵も接続もしていない場合



＊HIMEMのアドレスは、&HFC4Aと&HFC4B番地に格納されている。
　　HIMEM＝（&HFC4B）×&H100＋（&HFC4A）
※ドライブを内蔵または接続している場合でも、BASICのスタート時に SHIFT を押しながらスタートしたときは、この状態になります。

メモリマップ(MSX-DOS使用時)



*内蔵ドライブのみの場合。

MSX－DOSリファレンス表

MSX－DOSのファイルスペック

〈ドライブ名〉:〈ファイル名〉.〈拡張子〉　　例）Ａ：GAMEPROG．BAS

「″」で囲んで使用することはできない。
〈ドライブ名〉:を省略したときは、現在「プロンプト」（Ａ＞など）に表示されているドライブが指定される。
使用できる文字の種類、文字数、ワイルドカード、その他についてはＢＡＳＩＣと同じ。

ＭＳＸ－ＤＯＳの区切り記号

命令とそれに続く項目のあいだ、および各項目のあいだには必ず１文字分の空白を入れる。「，」や「;」は使用できない。

カレントドライブの切り替え

「コマンドライン」（現在カーソルが表示されているプロンプトに続く行）にドライブ名を入力して↵キーを押すことにより、プロンプトに表示されるドライブ名を変更することができる。

ＭＳＸ－ＤＯＳのディスクフォーマット

ＭＳＸ－ＤＯＳとＢＡＳＩＣのディスクのフォーマットは共通。したがって、どちらを使ってフォーマットしたディスクでも両方で使用することができる。ただしＭＳＸ－ＤＯＳのスタートには、ＭＳＸ－ＤＯＳのシステムファイルがコピーされているディスクが必要。

MSX－DOSの命令

書　式	機　能	文　例
ＢＡＳＩＣ	ＢＡＳＩＣに制御を移す	ＢＡＳＩＣ
ＣＯＰＹ 〈ファイルスペック1〉〈ファイルスペック2〉	ファイルをコピーする 別のディスクにコピーするときは､〈ファイルスペック2〉はドライブ名だけでも良い	ＣＯＰＹ　Ａ：ＧＡＭＥ．ＢＡＳ　Ｂ： ＣＯＰＹ　＊．＊　Ｂ：
ＤＡＴＥ [〈年〉 －/／ 〈月〉 －/／ 〈日〉]	[〈年月日〉]を省略して入力したときは、現在設定されている日付を表示し変更を求める。省略しないで入力したときは、表示なしで変更する。	ＤＡＴＥ　８９－２－１５
ＤＥＬ〈ファイルスペック〉	指定されたファイルをすべて消去する。〈ファイルスペック〉が「*.*」のときは、確認を求め、Ｙを入力すると実行する	ＤＥＬ　ＰＲＯＧ１ ＤＥＬ　＊．ＤＵＭ
ＤＩＲ [〈ファイルスペック〉] [/Ｐ] [/Ｗ]	ディスク内のファイルの情報の一覧表を表示する。 /Ｐはページモードの指定 /Ｗはファイル名のみの指定	ＤＩＲ ＤＩＲ　Ｂ：ＰＲＯＧ?．＊ ＤＩＲ　Ａ：／Ｐ ＤＩＲ　／Ｗ
ＥＲＡＳＥ〈ファイルスペック〉	ＤＥＬ命令と同じ。	ＥＲＡＳＥ　ＰＲＯＧ１．ＤＵＭ
ＦＯＲＭＡＴ [〈ドライブ名〉]	指定されたドライブのディスクをフォーマットする。	ＦＯＲＭＡＴ　Ｂ：
ＭＯＤＥ〈桁数〉	画面の１行の桁数を設定する（〈桁数〉は１～80の範囲）	ＭＯＤＥ　４０

書　　式	機　能	文　例
PAUSE [〈コメント〉]	バッチファイルで使用。 バッチファイルの実行を一時停止し、〈コメント〉の文字列とキー入力を求めるメッセージを表示する。このとき、CTRL ＋ C を入力するとバッチファイルの実行を中止する。それ以外のキーを押すと、実行を再開する	PAUSE　なにかキーをおして！
REM [〈コメント〉]	バッチファイルで使用。 〈コメント〉の文字列を表示。	REM　このままますすみます。
REN 〈旧ファイルスペック〉 〈新ファイルスペック〉	ファイル名を変更する 〈新ファイルスペック〉のドライブ名は無視される	REN　PROG1　PROG2
TIME [〈hh〉[:〈mm〉]]	時刻を表示し、変更を求める 〈hh〉:〈mm〉を入力したときは、表示なしで変更する 〈hh〉は00～23 〈mm〉は00～59	TIME　12：34
TYPE 〈ファイルスペック〉	アスキーファイルの内容を画面に表示する	TYPE　NIKKI．ASC

BASICエラーメッセージ表

※ 🖫 はドライブを内蔵または接続している場合のみ表示

メッセージ	エラーコード	説　　明
Bad allocation table 🖫	60	ディスクが正しくフォーマットされていない。
Bad drive name 🖫	62	ドライブ名の指定がまちがっている
Bad file mode 🖫	61	シーケンシャルファイルに対してPUT文、GET文またはLOF関数が使用されたり、ランダムファイルに対してLOAD文が使用された。またはOPEN文でモードの設定をまちがえている。
Bad file name	56	ファイル名が適切でない。
Bad file number	52	OPENしていない（開いていない）ファイル番号を指定した。指定したファイル番号がMAXFILES文で指定した数を越えている。
Can′t continue	17	CONT命令でプログラムの実行を再開できない。 エラーが発生して終了したときは再開できません。また、プログラムの一部を書き換えたり、CLEAR文などを実行すると、次に実行する文を示すポインタが壊れて再開できなくなります。
Device I/O error	19	データレコーダ、プリンタなどへの入出力中にエラーが起こった。
Direct statement in file	57	アスキー形式のファイルをロード中、ファイルの中に命令のコードが直接入っていた。
Disk full 🖫	66	ディスク内に空き領域が無い。
Disk I/O error 🖫	69	ディスク入出力操作において、回復できないエラーが発生した。
Disk offline 🖫	70	指定されたドライブにディスクが入っていない。 またはディスクがフォーマットされていない。
Disk write protected 🖫	68	ディスクのライトプロテクトタブが書き込み禁止の位置にあるのに、書き込みをともなう操作を行おうとした。
Division by zero	11	0による除算を行おうとした。または0に対して負の累乗を行おうとした。
FIELD overflow 🖫	50	FIELD文で定義したデータの長さの合計が、1レコードの大きさを越えた。（レコード長未定義のときは合計256バイト以内）
File already exists	65	NAME文などで指定された新しいファイル名のファイルが、すでに存在している。
File already open	54	すでに開いているファイルを再びOPENしようとした。または、同じファイル番号で開いたファイルをまだCLOSEしていない。 あるいは、開いているファイルをKILLしようとした。
File not found	53	指定されたファイルが見つからない。
File not OPEN	59	OPEN文で開いていないファイルに対して入出力を行おうとした。
File still open	64	ファイルのCLOSEを忘れている。
Illegal direct	12	ダイレクトモードでは使用できない命令をダイレクトモードで使おうとした。
Illegal function call	5	命令や関数の使いかた（数値の範囲や省略のしかたなど）がまちがっている。
Input past end	55	ファイルのデータをすべて読み込んだあと、さらにデータを入力しようとした。（ファイルのデータをすべて読み込んだかどうかは、EOF関数で調べてください。）
Internal error	51	BASIC内部でエラーが発生した。 （電源スイッチを入れ直してみてください）
Line buffer overflow	25	入力されたプログラムやデータが許された文字数を越えている。
Missing operand	24	入力された文の必要な項目が省略されている。
NEXT without FOR	1	NEXT文に対応するFOR文がない。
No RESUME	21	エラー処理ルーチンにRESUME文がない。

資料編

メッセージ	エラーコード	説　　明
Out of DATA	4	READ文で読み出すデータが不足している。
Out of memory	7	プログラムが長すぎたり、配列が大きすぎたりしてメモリが足りなくなった。
Out of string space	14	メモリの文字型変数用の領域が足りなくなった。 （文字列領域の大きさは、CLEAR文である程度調節可能）
Overflow	6	計算の結果や入力された数値が、型に応じた数値の範囲を越えた。
Redimensioned array	10	すでに定義した配列変数を再び定義しようとした。または定義せずに使用した配列変数を、あとから定義しようとした。
Rename across disk	71	異なるドライブのあいだでNAME文を実行させようとした。
RESUME without error	22	エラーが発生していないのにRESUME文を実行させようとした。またはエラーなしでエラー処理ルーチンを使おうとした。
RETURN without GOSUB	3	RETURN文に対応するGOSUB文がない。
Sequential I/O error	58	シーケンシャルファイルに対してランダム入出力を行おうとした。
String fomula too complex	16	指定した文字列が複雑すぎる。 （文字列を2つ以上に分けてください。）
String too long	15	文字列が 255文字より長い。
Subscript out of range	9	配列変数の添え字の値が、DIM文で定義した最大値を越えた。
Syntax error	2	文の書きかた（構文）をまちがえている。（かっこが正しく対応していない。命令や項目の綴りがちがう。区切り記号がないなど）
Too many files	67	指定されたディスク内のファイルの数が 112個に達しているときに、新しいファイルを作ろうとした。（1ディスクは 112ファイルまで）
Type mismatch	13	数式や数値型変数に文字列を使ったり、文字式や文字型変数に数値を使うなど、データの形式が合っていない。
Undefined line number	8	GOTO文、GOSUB文、IF文などで指定した行番号の行が、プログラム中に存在しない。
Undifined user function	18	DEF FN文で定義されていないユーザー定義サブルーチンを使おうとした。
Unprintable error	23 26～49 72～255	メッセージが定義されていないエラーが発生した。
Verify error	20	CLOAD？命令実行時のみ表示。 カセットに正しくセーブできていない。 （プログラムをセーブし直してください。）

MSX-DOSエラーメッセージ表

メッセージ	説　　明
Bad command or file name	命令かファイル名がまちがっている。
File creation error	新しいファイルの作成に失敗した。
File not found	指定されたファイルが見つからない。
Insufficient disk space	ディスク容量の空きがない。
Invalid date	入力された日付がまちがっている。
Invalid drive specification	ドライブ名の指定がまちがっている。
Invalid parameter	入力された文の項目がまちがっている。
Invalid time	入力された時刻がまちがっている。
Program too big to fit in memory	プログラムが大き過ぎてメモリに入りきらない。
Rename error	ファイル名の変更に失敗した。
Write error	ディスクへの書き込みに失敗しました。

ローマ字・かな変換表(BASIC使用時)

ＢＡＳＩＣの英数・かなモードで［SHIFT］を押しながら［かな］を押すと、ローマ字・かな変換でひらがなまたはカタカナを入力することができます。（ひらがな・カタカナは［CAPS］で切り替えます）

50音の対照

あ	A	い	I	う	U	え	E	お	O
か	KA	き	KI	く	KU	け	KE	こ	KO
さ	SA	し	SI SHI	す	SU	せ	SE	そ	SO
た	TA	ち	TI CHI	つ	TU TSU	て	TE	と	TO
な	NA	に	NI	ぬ	NU	ね	NE	の	NO
は	HA	ひ	HI	ふ	HU FU	へ	HE	ほ	HO
ま	MA	み	MI	む	MU	め	ME	も	MO
や	YA			ゆ	YU			よ	YO
ら	RA	り	RU	る	RU	れ	RE	ろ	RO
わ	WA			う	WU			を	WO
ん	NN								

濁音・半濁音の対照

か゛	GA	き゛	GI	く゛	GU	け゛	GE	こ゛	GO
さ゛	ZA	し゛	ZI JI	す゛	ZU	せ゛	ZE	そ゛	ZO
た゛	DA	ち゛	DI	つ゛	DU	て゛	DE	と゛	DO
は゛	BA	ひ゛	BI	ふ゛	BU	へ゛	BE	ほ゛	BO
は゜	PA	ひ゜	PI	ふ゜	PU	へ゜	PE	ほ゜	PO

拗音・促音の対照

		うぃ	WI			うぇ	WE		
きゃ	KYA	きぃ	KYI	きゅ	KYU	きぇ	KYE	きょ	KYO
き゛ゃ	GYA	き゛ぃ	GYI	き゛ゅ	GYU	き゛ぇ	GYE	き゛ょ	GYO
しゃ	SYA SHA	しぃ	SYI	しゅ	SYU SHU	しぇ	SYE SHE	しょ	SYO SHO
し゛ゃ	ZYA JYA JA	し゛ぃ	ZYI JYI	し゛ゅ	ZYU JYU JU	し゛ぇ	ZYE JYE JE	し゛ょ	ZYO JYO JO
ちゃ	TYA CYA CHA	ちぃ	TYI CYI	ちゅ	TYU CYU CHU	ちぇ	TYE CYE CHE	ちょ	TYO CYO CHO
ち゛ゃ	DYA	ち゛ぃ	DYI	ち゛ゅ	DYU	ち゛ぇ	DYE	ち゛ょ	DYO
てゃ	THA	てぃ	THI	てゅ	TYU	てぇ	TYE	てょ	TYO
て゛ゃ	DHA	て゛ぃ	DHI	て゛ゅ	DYU	て゛ぇ	DYE	て゛ょ	DYO
にゃ	NYA	にぃ	NYI	にゅ	NYU	にぇ	NYE	にょ	NYO
ひゃ	HYA	ひぃ	HYI	ひゅ	HYU	ひぇ	HYE	ひょ	HYO
ひ゛ゃ	BYA	ひ゛ぃ	BYI	ひ゛ゅ	BYU	ひ゛ぇ	BYE	ひ゛ょ	BYO
ひ゜ゃ	PYA	ひ゜ぃ	PYI	ひ゜ゅ	PYU	ひ゜ぇ	PYE	ひ゜ょ	PYO
ふぁ	FA	ふぃ	FI			ふぇ	FE	ふょ	FO
ふゃ	FYA	ふぃ	FYI	ふゅ	FYU	ふぇ	FYE	ふょ	FYO
みゃ	MYA	みぃ	MYI	みゅ	MYU	みぇ	MYE	みょ	MYO
りゃ	RYA	りぃ	RYI	りゅ	RYU	りぇ	RYE	りょ	RYO
う゛ぁ	VA	う゛ぃ	VI	う゛	VU	う゛ぇ	VE	う゛ぉ	VO

小文字の対照

ぁ	LA	ぃ	LI	ぅ	LU	ぇ	LE	ぉ	LO
ゃ	LYA	ぃ	LYI	ゅ	LYU	ぇ	LYE	ょ	LYO

※「っ」は同じ子音を2回押します。
※「X」を押すと「ー」（長音）が表示されます。
※この表は、英数・かなモード時に変換される文字をあらわしています。漢字モードでも、半角文字の入力時は同じです。
※漢字モードの全角文字入力時でのローマ字・かな変換は、次の点が異なります。

・「゛」と「゜」は前の文字と合わせて1文字になります。　例）「が」、「ぱ」
・全角文字には「ヴ」はありますが、「う゛」がありません。「う」と「゛」の2文字で入力してください。
・記号は次の通りになります。

"／"（スラッシュ）　…………………… "・"（中点）
"．"（ピリオド）　…………………… "。"（句点）
"，"（カンマ）　…………………… "、"（読点）
SHIFT ＋"．"　…………………… "．"（ピリオド）
SHIFT ＋"，"　…………………… "，"（カンマ）
［（左角かっこ）　…………………… "「"（左かぎかっこ）
］（右角かっこ）　…………………… "」"（右かぎかっこ）

命令索引

BASICとMSX－DOSの命令、関数、システム変数などの索引です。

資料編

資料編

推奨図書一覧

ＭＳＸホームコンピュータ読本	（アスキー）
新・ＭＳＸの基礎知識	（アスキー）
ＭＳＸ２パーソナルユースのすべて	（アスキー）
ＭＳＸ２ ＢＡＳＩＣ 入門	（アスキー）
ＭＳＸビギナーズＢＡＳＩＣ	（アスキー）
ＭＳＸビギナーズ・ハンドブック	（エム・アイ・エー）
ＢＡＳＩＣのコツ	（アスキー）
エラー撃退ミニ事典	（アスキー）
ＢＡＳＩＣゲーム教室	（アスキー）
ＭＳＸ ＢＡＳＩＣゲーム集①	（エム・アイ・エー）
ＭＳＸ ＢＡＳＩＣゲーム集②	（エム・アイ・エー）
ＭＳＸ ＢＡＳＩＣゲーム集③	（エム・アイ・エー）
プログラム Ｄ．Ｊ．	（アスキー）
面白パズルブック	（アスキー）
アニメ Ｃ．Ｇ．に挑戦	（アスキー）
マイコン・ジュークボックス	（アスキー）
トランプゲーム集	（アスキー）
音楽のおくりもの	（アスキー）
ことばの実験室	（アスキー）
アクションゲーム３８	（アスキー）
知能ゲーム３８	（アスキー）
ＭＳＸ２テクニカルハンドブック	（アスキー）

漢字コード表の補足説明

パソコンの取扱説明書や内蔵ソフト使用説明書などの「ＪＩＳ第１水準漢字コード表」は、主に内蔵ソフトや付属ソフトで使用するよう作られているため、一部の記号などのコードがＢＡＳＩＣとは異なります。
ＢＡＳＩＣでＪＩＳ非漢字（ひらがな、カタカナ、英数字、ギリシャ文字、ロシア文字、記号など）を使用するときは、下の表を参照してください。

ＪＩＳコードは縦軸の値（Ａ）と横軸の（Ｂ）との和となります。
例）●のコード＝2160＋1C＝217C　　（数値はいずれも16進数）

	A＼B	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20
記号	2120		、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨	＾	￣	＿	ヽ	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―	‐	／	＼
	2140	～	∥	｜	…	‥	‘	’	“	”	（	）	〔	〕	［	］	｛	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×	÷
	2160	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	°	′	″	℃	￥	＄	￠	￡	％	＃	＆	＊	＠	§	☆	★	○	●	◎	◇		
	2220	◆	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	〓	㈱	㈲	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	g	m	cm	mm	km	cm²
	2240	m²	ml	mg	kg	us	㈹	KK	TEL	No.	インチ		グラム	メートル	センチ	ミリ	ヘクタール	アール	ドル	トン	キロ	キログラム	キロワット	ワット	リットル	パーセント	ページ	ヤード	フィート	①	②	③	④
	2260	⑤	⑥	⑦	⑧	⑨	⑩	【	】	〶	☎	🗨	∫	√	￣	┌┐	││	└┘	⌒	◡	［	＝	］	（	）	CR1	CR2	㊡	時	⇧	⇩		
英数字	2320	◀	◀	‥	▪▪	▪	：	⇨									０	１	２	３	４	５	６	７	８	９							
	2340	Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ	Ｎ	Ｏ	Ｐ	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ	Ｖ	Ｗ	Ｘ	Ｙ	Ｚ						
	2360	ａ	ｂ	ｃ	ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ	ｋ	ｌ	ｍ	ｎ	ｏ	ｐ	ｑ	ｒ	ｓ	ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ						
ひらがな	2420	ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た	だ
	2440	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み	む
	2460	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ	ゐ	ゑ	を	ん													
カタカナ	2520	ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ	ダ
	2540	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ	ム
	2560	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ										
ギリシャ文字	2620	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω								
	2640	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω								
ロシア文字	2720	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю
	2740	Я															а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н	о
	2760	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я															

資料編

松下電器産業株式会社 情報機器部
〒571 大阪府門真市大字門真1006 電話(06)908-1151

DFQF2202ZA
S1088-0